Lecture Notes in Artificial Intelligence 2123

Subseries of Lecture Notes in Computer Science
Edited by J. G. Carbonell and J. Siekmann

Lecture Notes in Computer Science
Edited by G. Goos, J. Hartmanis and J. van Leeuwen

Springer
Berlin
Heidelberg
New York
Barcelona
Hong Kong
London
Milan
Paris
Singapore
Tokyo

Petra Perner (Ed.)

Machine Learning and Data Mining in Pattern Recognition

Second International Workshop, MLDM 2001
Leipzig, Germany, July 25-27, 2001
Proceedings

Springer

Series Editors

Jaime G. Carbonell, Carnegie Mellon University, Pittsburgh, PA, USA
Jörg Siekmann, University of Saarland, Saarbrücken, Germany

Volume Editor

Petra Perner
Institute of Computer Vision and Applied Computer Sciences
Arno-Nitzsche-Str. 45, 04277 Leipzig, Germany
E-mail: ibaiperner@aol.com

Cataloging-in-Publication Data applied for

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

Machine learning and data mining in pattern recognition : second international workshop ; proceedings / MLDM 2001, Leipzig, Germany, July 25 - 27, 2001. Petra Perner (ed.). - Berlin ; Heidelberg ; New York ; Barcelona ; Hong Kong ; London ; Milan ; Paris ; Singapore ; Tokyo : Springer, 2001
(Lecture notes in computer science ; 2123 : Lecture notes in artificial intelligence)
ISBN 3-540-42359-1

CR Subject Classification (1998): I.2, I.5, I.4, F.4.1, H.3

ISBN 3-540-42359-1 Springer-Verlag Berlin Heidelberg New York

Springer-Verlag Berlin Heidelberg New York
a member of BertelsmannSpringer Science+Business Media GmbH

http://www.springer.de

Printed in Germany

Typesetting: Camera-ready by author, data conversion by Steingräber Satztechnik GmbH, Heidelberg
Printed on acid-free paper SPIN: 10839922 06/3142 5 4 3 2 1 0

Preface

The papers published in this book were presented at the second workshop on Machine Learning and Data Mining in Pattern Recognition MLDM. Those of you familiar with the first workshop will notice that the ideas related to this topic are spreading to many researchers. We received several excellent papers on subjects ranging from basic research to application oriented research. The subjects are Case-Based Reasoning, Rule Induction, Grammars, Clustering, Data Mining on Multimedia Data, Content-Based Image Retrieval, Statistical and Evolutionary Learning, Neural Networks, and Learning for Handwriting Recognition. The whole spectrum of topics in MLDM was represented at the workshop with emphasis on images, text, and signals, and temporal spatial data. We also took a step in the direction of our decision made at the last TC3 Machine Learning meeting in Barcelona, to introduce the field to researchers outside the computer science or pattern recognition community. We welcomed medics, specialists of data base marketing, and mechanical engineers to our workshop. These researchers reported their experience and the problems they have in applying Data Mining. By sharing their experience with us they give new impulses to our work.

The workshop was organized by the Leipzig Institute of Computer Vision and Applied Computer Sciences. Many thanks to Maria Petrou for co-chairing MLDM 2001 with me.

It is my pleasure to thank the invited speakers for accepting our invitation to give lectures and contribute papers to the proceedings. I would also like to express my appreciation to the reviewers for their precise and highly professional work. I appreciate the help and understanding of the editorial staff at Springer-Verlag, and in particular Alfred Hofmann, who supported the publication of these proceedings in the LNAI series.

Last but not least, I wish to thank all the speakers and participants for their interest in this workshop. I hope you enjoyed the workshop and that you will return to present your new ideas at MLDM 2003.

July 2001 Petra Perner

IAPR International Workshop on Machine Learning and Data Mining in Pattern Recognition

Co-chairs

Petra Perner
IBaI Leipzig / Germany

Maria Petrou
University of Surrey / United Kingdom

The Aim of the Workshop

The aim of the workshop was to bring together researchers from all over the world dealing with machine learning and data mining in order to discuss the recent status of the research and to direct further developments. All kinds of application were welcome. Special preference was given to multimedia related applications.
It was the second workshop in a series of workshops dealing with this specific topic. The first workshop was published in *P. Perner and M. Petrou, Machine Learning and Data Mining in Pattern Recognition MLDM99, LNAI 1715, Springer Verlag 1999* and in a special issue of *Pattern Recognition Letters.*
The topics covered include:

- inductive learning including decision trees
- rule induction learning
- conceptual learning
- case-based learning
- statistical learning
- neural net based learning
- organisational learning
- evolutional learning
- probabilistic information retrieval

Applications include but are not limited to medical, industrial, and biological applications.

Researchers from the machine learning community were invited to present new topics in learning, pertinent to our research field.

Table of Contents

Invited Paper

Case-Based Reasoning and Associative Memory

Rule Induction and Grammars

Clustering and Conceptual Clustering

Data Mining on Signal, Images, Text and Temporal-Spatial Data

Nonlinear Function Learning and Neural Net Based Learning

Learning for Handwriting Recognition

Statistical and Evolutionary Learning

Technology of Text Mining

Ari Visa

Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
Ari.Visa@tut.fi

Abstract. A large amount of information is stored in databases, in intranets or in Internet. This information is organised in documents or in text documents. The difference depends on the fact if pictures, tables, figures, and formulas are included or not. The common problem is to find the desired piece of information, a trend, or an undiscovered pattern from these sources. The problem is not a new one. Traditionally the problem has been considered under the title of information seeking, this means the science how to find a book in the library. Traditionally the problem has been solved either by classifying and accessing documents by Dewey Decimal Classification system or by giving a number of characteristic keywords. The problem is that nowadays there are lots of unclassified documents in company databases and in intranet or in Internet.
First one defines some terms. Text filtering means an information seeking process in which documents are selected from a dynamic text stream. Text mining is a process of analysing text to extract information from it for particular purposes. Text categorisation means the process of clustering similar documents from a large document set. All these terms have a certain degree of overlapping.
Text mining, also know as document information mining, text data mining, or knowledge discovery in textual databases is an merging technology for analysing large collections of unstructured documents for the purposes of extracting interesting and non-trivial patterns or knowledge. Typical subproblems that have been solved are language identification, feature selection/extraction, clustering, natural language processing, summarisation, categorisation, search, indexing, and visualisation. These subproblems are discussed in detail and the most common approaches are given.
Finally some examples of current uses of text mining are given and some potential application areas are mentioned.

1 Introduction

Nowadays a large amount of information is stored in intranet, internet or in databases. Customer comments and communications, trade publications, internal research reports and competitor web sites are just a few examples of available electronic data. The access to this information is many times organized through the World Wide Web. There are already some commercial tools available that

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 1–11, 2001.

are defined as knowledge solutions. The reason is clear; everyone needs a solution for handling the large volume of unstructured information. This is either in intuitive way clear but before a more detailed discussion it is useful to define some phrases and concepts.

We should keep in mind the distinction between data, information and knowledge. These terms can be defined in several ways but the following definitions are useful in Data and Text Mining purposes. The lowest level, data, used to be clear. It is a measurement, a poll, a simple observation. The next level, information, is already more diffuse. It is an observation based on data. We have for instance noticed a cluster among the data or a relation between data items. The highest level, knowledge, is the most demanding. It can be understood as a model or a rule. We know from theory of science that lots of careful planning and experimentation are needed before we can state to know something, we have knowledge.

The phrase document is a more complicated term. It is clear that work gets done through documents. When a negotiation draws to a close, a document is drawn up, an accord, a law, a contract, an agreement. When research culminates, a document is created and published. The knowledge is transmitted through documents: research journals, text books and newspapers. Documents are information and knowledge organized and presented for human understanding. A typical document of today is either printed or electrical one. The printed documents are transferable to electrical ones by optical scanning and Optical Character Recognition (OCR) methods. Tables, figures, graphics, and pictures are problematic under this transform process. The electrical documents are either hierarchical or free. The hierarchical documents use some kind of page description language (PDL), for instance Latex, and imager programs, which take PDL representations to a printable or projectable image. Free documents may contain only free text or free text with tables, figures, graphics, and pictures. Besides the mentioned document types two new types are coming popular: multimedia documents with voice and video in addition to text and pictures, and hyper-media documents that are non-linear documents. In the continuation one concentrates mainly on free text without any insertions as, tables, figures, graphics, pictures, and so on.

The need to manage knowledge and information is not a new one. It has existed as long as the mankind or at least the libraries have existed. Roughly we can say that the key questions are how to store information, how to find it and how to display it. Now we concentrate on the information seeking [3]. First it is useful to overview different kinds of information seeking processes, see Table 1. The presentations are general but we concentrate on the electrical form existing documents. Please, keep in mind that any information seeking process begins with the users' goal. Firstly, information filtering systems are typically designed to sort through large volumes of dynamically generated information and present the user with sources of information that are likely to satisfy his or her information requirement. By information source we mean entities which contain information in a form that can be interpreted by the user. The information

filtering system may either provide these entities directly, or it may provide the user with references to the entities The distinguishing features of the information filtering process are that the users' information needs are relatively specific, and that those interests change relatively slowly with respect to the rate at which information sources become available. Secondly, a traditional information retrieval system can be used to perform an information filtering process by repeatedly accumulating newly arrived documents for a short period, issuing an unchanging query against those documents, and then flushing the unselected documents. Thirdly, another familiar process is the process of retrieving information from a database. The distinguishing feature of the database retrieval process is that the output will be information, while in information filtering, the output is a set of entities (e.g. documents) which contain the information which is sought. For example, using an library catalog to find the title of a book would be a database access process. Using the same system to discover whether any new books about a particular topic have been added to the collection would be an information filtering process. Fourthly, the information extraction process is similar to the database access in that the goal is to provide information to the user, rather than entities which contain information. In the database access process information is obtained from some type of database, while in information extraction the information is less well structured (e.g. the body of an electronic mail message). Fifthly, one variation on the information extraction and database access processes is what is commonly referred to as alerting. In the alerting process the information need is assumed to be relatively stable with respect to the rate at which the information itself is changing. Monitoring an electronic mailbox and alerting the user whenever mail from a specific user arrives is one example of an information alerting process. Sixthly, browsing can be performed on either static or dynamic information sources, browsing has aspects similar to both information filtering and information retrieval. Surfing the World Wide Web is an example of browsing relatively static information, while reading an online newspaper would be an example of browsing dynamic information. The distinguishing feature of browsing is that the users' interests are assumed to be broader than in the information filtering or retrieval processes. Finally, there is a case when one tumbles over an interesting piece of information.

According to ANSI 1968 Standard (American National Standards Institutes, 1968), an index is a systematic guide to items contained in, or concepts derived from, a collection. These items or derived concepts are represented by entities in a known or stated searchable order, such as alphabetical, chronological, or numerical. Indexing is the process of analysing the informational content of records of knowledge and expressing the information content in the language of the indexing system. It involves selecting indexable concepts in a document and expressing these concepts in the language of the indexing system (as index entries) and an ordered list.

Natural Language processing [20] is a broad topic and an important topic for Text Data Mining but here I give only some terms. Stemming is a widely used method for collapsing together different words with a common stem [16]. For

Table 1. Examples of different information seeking processes.

Process	Information Need	Information Sources
Information Filtering	Stable and Specific	Dynamic and Unstructured
Information Retrieval	Dynamic and Specific	Stable and Unstructured
Database Access	Dynamic and Specific	Stable and Structured
Information Extraction	Specific	Unstructured
Alerting	Stable and Specific	Dynamic
Browsing	Broad	Unspecified
By Random Search	Unspecified	Unspecified

instance, if a text includes words Marx, Marxist, and Marxism, it is reasonable to observe the distribution of the common stem Marx instead of three separate distributions of these words. Accordingly, synonymy, hyponymy, hypernymy, and other lexical relatedness of words are detected by using thesauruses or techniques that define semantic networks of words.

Information or in this case text categorisation requires that there are existing categories. There have been several approaches but nowadays in libraries books are classified and accessed according to Dewey Decimal Classification (DCC) [7,6,9]. DCC defines a set of 10 main classes of documents that cover all possible subjects a document could be referring to. Each class is then divided into ten divisions and each division is divided into ten sections. In cases when we do not have existing categories we talk about text clustering. We collect similar documents together, the similarity is defined by a measure.

Data Mining contains the metaphor of extracting ore from rock. In practice Data Mining refers to finding patterns across large datasets and discovering heretofore unknown information. In the same way, as Data Mining can not be accessing data bases, Text Mining can not be finding documents. The emphasis in Information Retrieval is in finding document. The finding patterns in text collections is exactly what has been done in Computational Linguistics. [20]. Text mining, also know as document information mining, text data mining, or knowledge discovery in textual databases is an merging technology for analysing large collections of unstructured documents for the purposes of extracting interesting and non-trivial patterns or knowledge [12]. The aim is to extract heretofore undiscovered information from large text collections.

2 Text Mining Technology

Keeping in mind the evolution to Data Mining and to Text Mining one can state that there are need for tools. In generally tools for clustering, visualisation and interpretation are needed. For instance, under the document exploration tools to organise the documents and to navigate through a large collection of documents are needed. Typical technologies are text categorisation, text clustering, summarisation, and visualisation. Under the text analysis and domain-specific knowledge discovery technologies as question answering, text understanding, in-

formation extraction, prediction, associative discovery, and trend analysis are adequate. I will review most important steps and techniques in Text Mining.

In text mining, as in data mining, there are some initial problems before the work itself can be started. Firstly, some data pre-processing is needed [20]. The key idea is to transform the data into such a form that it can be processed. It might be removal of pictures, tables or text formation or it might contain replacement of mathematical symbols, numbers, URLs, and email addresses with special dummy tokens. If OCR techniques are used the pre- processing step may consist spelling checking. However, during the pre-processing stage some caution is needed, hence it is easy to destroy some structural information. The pre-processing might also be a language related process. In languages as Germany, Finnish, or Russian stemming might be needed. The information of the actual language of the text is very useful. The processing of mono linguistic collection is more straight forward than cross language or the multilingual processing [27,21,22].

It is common that long documents are summarised. The first attempts were made in the 1950's, in the form of Luhn's auto-extracts [19], but unfortunately since then there has been little progress. The reason is easy to understand by defining a summary. A summary text is a derivative of a source text condensed by selection or generalisation on important content. This broad definition includes a wide range of specific variations. Summarising is conditioned by input factors categorising source form and subject, by purpose factors referring to audience and function, and by output factors including summary format and style. The main approaches have been the following: Source text extraction using statistical cues to select key sentences to form summaries [24]. Approaches using scripts or frames to achieve deeper representations and an explicitly domain-oriented kind motivated properties of the world [35]. There has been research combining different information types in presentation. Thus combines linguistic theme and domain structure in source representations, and seeks salient concepts in these for summaries [8].

After the possible summarisation and the pre-processing some kind of encoding is needed. The key questions is the representation of text documents. This question is closely related to feature selection but here the term feature has a broader meaning than in pattern recognition. There are the two main approaches: use of index terms or the use of free text. These approaches are not competing each other but completing. It is common that natural language processing is used to reach index terms. It is possible to proceed directly with index terms and Boolean algebra as one does in information retrieval systems with queries. This is known as Boolean model [2]. The model is binary, the frequency of term has no effect. Due to its uncomplicated semantics and straightforward calculation of results using set operations, the Boolean model is widely used e.g. in commercial search tools. The vector space model is introduced by Salton [29,28] encode documents in a way suitable for fast distance calculations. Each document is represented as a vector in a space, where the dimension is equal to the number of terms in vocabulary. In this model the problem of finding suit-

able documents to a query becomes that of finding the closest document vectors for a query vector, either in terms of distance or of angle. Vector space models underlie the research in many modern information retrieval systems. The probabilistic retrieval model makes explicit the Probability Ranking Principle that can be seen underlying most of the current information retrieval research [20]. For a given query, estimate the probability that a document belongs to the set of relevant documents and return documents in the order decreasing probability of relevance. The key question is, how to obtain the estimates regarding which documents are relevant to a given query. These simple search approaches are similar to association and associative memories. The method is to describe a document with index terms and to build a connection between the index terms and the document. To build this connecting function among other methods artificial neural networks have been used [14].

The use of free text is more demanding. Instead of using index terms it is possible to use other features to represent a document. A common approach is to view a document as a container of words. This is called bag-of-words encoding. It ignores the order of the words as well as any punctuation or structural information, but retains the number of times each word appears. The idea is based on the work of Zipf [36] and Luhn [19]. The famous constant rank-frequency law of Zipf states that if the word frequencies are multiplied by their rank order (i.e. the order of their frequency of occurrence), the product is approximately constant. Luhn remarks that medium-frequency words are most significant. The most frequent words (the, of, and, etc.) are least content-bearing, and the least frequent words are usually not essential for the content of a document either. A straightforward numeric representation for the bag of words-model is to present documents in the vector space model, as points in a t-dimensional Euclidean space where each dimension corresponds to a word of a vocabulary. The i:th component di of the document vector express the number of times the word with index i occurs in the document. The described method is called term frequency document. Furthermore, each word may have an associated weight to describe its significance. This is called term weighting, The similarity between two documents is defined either as the distance between the points or as the angle between the vectors. To consider only the angle discards the effect of document length. Another way to eliminate the influence of document length is to use inverse document frequency this means that the term frequency is normed with document frequency. A variation of the inverse document frequency is the residual inverse document frequency is defined as the difference between the logs of the actual inverse document frequency and inverse document frequency predicted by Poisson model. Another main approach is term distributions models. They assume that the occurrence frequency of words obeys a certain distribution. Common models are the Poisson model, the two-Poisson model, and the K mixture model. The third main approach is to consider the relationships between words. A term-by-document matrix that can be deducted from their occurrence patterns across documents. This notation is used in a method called Latent Semantic Indexing [20], which applies singular-value decomposition to the document-by-word ma-

trix to obtain a projection of both documents and words into a space referred as the latent space. Dimensionality reduction is achieved by retaining only the latent variables with the largest variance. Subsequent distance calculations between documents or terms are then performed in the reduced-dimensional latent space.

The feature selection which means developing richer models that are computationally feasible and possible to estimate from actual data remains a challenging problem. However, facing this challenge is necessary if harder tasks related e.g. to language understanding and generation are to be tackled.

When we have produced either suitable models or gathered the features we are ready to the next step, clustering. Clustering algorithms partition a set of objects into groups or clusters. The methods are principally the same as in Data Mining, but the popularity of algorithms vary [20]. The clustering is one of the most important steps in Text Mining. The main types of clustering are hierarchical and non- hierarchical. The tree of a hierarchical clustering can be produced either bottom-up, by starting with the individual objects and grouping the most similar ones, or top-down, whereby one starts with all the objects and divides them into groups so as to maximise within-group similarity. The commonly used similarity functions are single-link, complete link, and group-average. The similarity between two most similar, or two least similar and average similarity between members is calculated. Non-hierarchical algorithms often start out with a partition based on randomly selected seeds (usually one seed per cluster), and then refine this initial partition. Most non-hierarchical algorithms employ several passes of reallocating objects to the currently best cluster whereas hierarchical algorithms need only one pass. A typical non-hierarchical algorithm is K-means that defines clusters by the centre of mass of their members. We need a set of initial cluster centres in the beginning. Then we go through several iterations of assigning each object to the cluster whose centre is closest. After all objects have been assigned, we recomputed the centre of each cluster as the centroid or mean of its members. The distance function is Euclidean distance [20]. In some case we also view clustering as estimating a mixture of probability distributions. In those cases we use EM algorithm [20]. The EM algorithm is an iterative solution to the following circular statements: Estimate: If we knew the value of a set of parameters we could compute the expected values of the hidden structure of the model. Maximize: If we knew the expected values of the hidden structure of the model, then we could compute the maximum likelihood value of a set of parameters.

Depending on the research task some text segmentation may be needed. It is also called information extraction. In information extraction also known as message understanding, unrestricted texts are analysed and a limited range of key pieces of task specific information are extracted from them. The problem is many times how to break documents into topically coherent multi-paragraph subparts. The basic idea of the algorithm is to search for parts of a text where the vocabulary shifts from one subtopic to another. These points are then interpreted as the boundaries of multi-paragraph units [13].

Finally, it is still important to visualise the features, the clusters, or the documents. Quantitative information has been presented using graphical means [32] since 1980s but during 1990s the scientific visualisation has developed a lot. This development helps us in information seeking, and in document exploration an in management. Quite a lot of has been done in connection to the project Digital Library. Properties of large sets of textual items, e.g., words, concepts, topics, or documents, can be visualised using one-, two-, or three- dimensional spaces, or networks and trees of interconnected objects, dendrograms [31]. Semantic similarity and other semantic relationships between large numbers of text items have usually been displayed using proximity. Some examples of that are the Spire text engine [34], document maps organised with Self Organized Map (SOM) [26,18,17,30], using coloured arcs in Rainbows [10], and with colour coordination of themes in the ET-map [23]. Another approach is to use the visual metaphor of natural terrain that has been used in visualising document density and clustering in ThemeView [34], in WEBSOM [15], and in a map of astronomical texts [25]. Relationships, e.g. influence diagrams between scientific articles, have been constructed based on citations and subsequently visualised as trees or graphs in BibRelEx [5]. Citeseer [4] offers a browsing tool for exploring networks of scientific articles through citations as well as both citation- and text-based similarity between individual articles. Searching is used to obtain a suitable starting-point for browsing. Term distributions within documents retrieved by a search engine have been visualised using TileBars [11]. Visualisation is rapidly developing field also in Text Mining.

3 Some Applications

I introduce briefly three applications of Text Mining.

In the first case we treated the annual reports that contained information both in numerical and in textual form [1]. More and more companies provide their information in electronic form this is the reason why this approach was selected. The numerical data was treated by Data Mining and the textual data by Text Mining. A multi level approach based on word, sentence, and paragraph levels were applied. The interesting point was to find out that the authors seems to emphasis the results even though the numerical facts are not supporting their attitude.

In the second case Text Mining has been used to identify the author [33]. A multi level approach based on word and sentence levels has been applied on database containing novels and poems. For authorship attribution purposes the authors William Shakespeare, Edgar Allan Poe, and George Bernard Shaw were selected. The interesting point was to identify and separate the authors.

In the third case a different approach is taken. In this approach WEBSOM [15] is used to visualise a database with 7 million patent abstracts. This is an typical exploration example and the map offers additional information regarding the results that cannot be conveyed by the one-dimensional list of results.

4 Discussion

Text mining is best suited for discovery purposes, learning and discovering information that was previously unknown. Some examples how text mining are used are: exploring how market is evolving, or looking for new ideas or relations in topics. While a valuable tool, text mining is not suited to all purposes. Just as you would not use data mining technology to do a simple query of your database, text mining is not the most efficient way to isolate a single fact. Text mining is only a support tool. However, text mining is relevant because of the enormous amount of knowledge, either within an organisation or outside of it. The whole collection of text is simply too large to read and analyse easily. Furthermore, it changes constantly and requires ongoing review and analysis if one is to stay current. A text mining product supports and enhances the knowledge worker's creativity and innovation with open-ended exploration and discovery. The individual applies intelligence and creativity to bring meaning and relevance to information, turning information into knowledge. Text mining advances this process, empowering the knowledge worker to explore and gain knowledge from the knowledge base. The text mining delivers the best results when used with information that meets the following criteria: The information must be textual. Numerical data residing within a database structure are best served by existing data mining technologies. The value of text mining is directly proportional to the value of the data you are mining. The more important the knowledge contained in the text collection, the more value you will derive by mining the data. The content should be explicitly stated within the text. Scientific and technical information are good examples of explicitly stated material. It seems that highly structured information already resides within a navigable organisation. Text mining is not as valuable in those cases, provided the structure of the information makes some sense. Text Mining is most useful for unorganised bodies of information, particularly those that have an ongoing accumulation and change. Bodies of text that accumulate chronologically are typically unorganised, and therefore good candidates for text mining.

There are already some commercial techniques and tools for text mining purposes. However, the text mining field is rapidly evolving, the following will guide users in what to consider when selecting among text mining solutions. One should consider the requirements of manual categorisation, tagging or building of thesauri. It is useful if long, labor-intensive integrations are avoided. The automatic identification and indexing of concepts within the text will also save a great deal of work. It is also nice if the tool can present visually a high level view of the entire scope of the text, with the ability to quickly drill down to relevant details. It is also nice if the tool enables users to make new association and relationships, presenting paths for innovation, or exploration and integrates with popular collaborative workflow solutions. Finally, if the tool scales to process any size data set quickly and it handles all types of unstructured data formats and runs on multiple formats.

Acknowledgments

The financial support of TEKES (grant number 40943/99) is gratefully acknowledged.

References

1. B. Back, J. Toivonen, H. Vanharanta, and A. Visa. Toward Computer Aided Analysis of Text. *The Journal of The Economic Society of Finland*, 54(1):39–47, 2001.
2. R. Baeza-Yates and B. Ribeiro-Neto, editors. *Modern Information Retrieval*. Addison Wesley Longman, 1999.
3. D. C. Blair. *Language and Representation in Information Retrieval*. Elsevier, Amsterdam, 1990.
4. K. Bollacker, S. Lawrence, and C. L. Giles. Citeseer: An autonomous web agent for automatic retrieval and identification of interesting publications. In *Proceedings of 2nd International ACM Conference on Autonomous Agents*, pages 116–123. ACM Press, 1998.
5. A. Brüggemann-Klein, R. Klein, and B. Landgraf. BibRelEx – Exploring Bibliographic Databases by Visualization of Annotated Content-Based Relations. *D-Lib Magazine*, 5(11), Nov. 1999.
6. M. Dewey. *A Classification and subject index for cataloguing and arranging the books and pamphlets of a library*. Case, Lockwood & Brainard Co., Amherst, MA, USA, 1876.
7. M. Dewey. Catalogs and Cataloguing: A Decimal Classification and Subject Index. In *U.S. Bureau of Education Special Report on Public Libraries Part I*, pages 623–648. U.S.G.P.O., Washington DC, USA, 1876.
8. U. Hahn. Topic parsing: accounting for text macro structures in full-text analysis. *Information Processing and Management*, 26(1):135–170, 1990.
9. S. P. Harter. *Online Information Retrieval*. Academic Press, Orlando, Florida, USA, 1986.
10. S. Havre, B. Hetzler, and L. Nowell. ThemeRiver^{TM}: In search of trends, patterns, and relationships. In *Proceedings of IEEE Symposium on Information Visualization (InfoVis'99)*, San Francisco, CA, USA, Oct. 1999.
11. M. A. Hearst. TileBars: Visualization of Term Distribution Information in Full Text Information Access. In *Proceedings of the ACM Conference on Human Factors in Computing Systems, (CHI'95)*, pages 56–66, 1995.
12. M. A. Hearst. Untangling text data mining. In *Proceedings of ACL'99, the 37th Annual Meeting of the Association for Computational Linguistics*, June 1999.
13. M. A. Hearst and C. Plaunt. Subtopic Structuring for Full-Length Document Access. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 59–68, 1993.
14. V. J. Hodge and J. Austin. An evaluation of standard retrieval algorithms and a binary neural approach. *Neural Networks*, 14(3):287–303, Apr. 2001.
15. T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, J. Honkela, V. Paatero, and A. Saarela. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 11(3):574–585, May 2000.
16. T. Lahtinen. *Automatic indexing: an approach using an index term corpus and combining linguistic and statistical methods*. PhD thesis, Department of General Linguistics, University of Helsinki, Finland, 2000.

17. X. Lin. Map displays for information retrieval. *Journal of the American Society for Information Science*, 48(1):40–54, 1997.
18. X. Lin, D. Soergel, and G. Marchionini. A Self-Organizing Semantic Map for Information Retrieval. In *Proceedings of 14th Annual International ACM/SIGIR Conference on Research & Development in Information Retrieval*, pages 262–269, 1991.
19. H. P. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.
20. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing*. The MIT Press, Cambridge, Massachusetts, USA, 1999.
21. P. Nelson. Breaching the language barrier: Experimentation with Japanese to English machine translation. In D. I. Raitt, editor, *15th International Online Information Meeting Proceedings*, pages 21–33. Learned Information, Dec. 1991.
22. D. W. Oard and B. J. Dorr. A Survey of Multilingual Text Retrieval. Technical Report CS-TR-3615, University of Maryland, 1996.
23. R. Orwig, H. Chen, and J. F. Nunamaker. A Graphical, Self-Organizing Approach to Classifying Electronic Meeting Output. *Journal of the American Society for Information Science*, 48(2):157–170, 1997.
24. C. Paice. Constructing literature abstracts by computer: techniques and prospects. *Information Processing and Management*, 26(1):171–186, 1990.
25. P. Poinçot, S. Lesteven, and F. Murtagh. A spatial user interface to the astronomical literature. *Astronomy and Astrophysics Supplement Series*, 130:183–191, 1998.
26. H. Ritter and T. Kohonen. Self-Organizing Semantic Maps. *Biological Cybernetics*, 61(4):241–254, 1989.
27. G. Salton. Automatic processing of foreign language documents. *Journal of the American Society for Information Science*, 21(3):187–194, 1970.
28. G. Salton. *Automatic Text Processing*. Addison-Wesley, 1989.
29. G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
30. J. C. Scholtes. *Neural Networks in Natural Language Processing and Information Retrieval*. PhD thesis, Universiteit van Amsterdam, Amsterdam, Netherlands, 1993.
31. B. Shneiderman. The Eyes Have It: A Task by Data Type Taxonomy for Information Visualizations. In *Proceedings of IEEE Symposium on Visual Languages, (VL)*, pages 336–343, Sept. 1996.
32. E. R. Tufte. *The Visual Display of Quantitative Information*. Graphic Press, 1983.
33. A. Visa, J. Toivonen, S. Autio, J. Mäkinen, H. Vanharanta, and B. Back. Data Mining of Text as a Tool in Authorship Attribution. In B. V. Dasarathy, editor, *Proceedings of AeroSense 2001, SPIE 15th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls. Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, volume 4384, Orlando, Florida, USA, Apr. 16–20 2001.
34. J. A. Wise. The Ecological Approach to Text Visualization. *Journal of the American Society of Information Science*, 50(13):1224–1233, 1999.
35. S. R. Young and P. J. Hayes. Automatic classification and summarisation of banking telexes. In *Proceedings of The Second Conference on Artificial Intelligence Applications*, pages 402–408, 1985.
36. G. K. Zipf. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, Massachusetts, USA, 1949.

Evaluation of Clinical Relevance of Clinical Laboratory Investigations by Data Mining

Ulrich Sack and Manja Kamprad

Institute of Clinical Immunology and Transfusion Medicine
Max Bürger Research Center
Johannisallee 30, 04103 Leipzig, Germany
mail@ulrichsack.de

Abstract. The diagnostic investigation of immunologically influenced diseases includes the determination of serological and cellular parameters in the peripheral blood of patients. For the detection of these parameters, a variety of well established and new fashioned immunoassays are available. Since these test kits have been shown to yield highly different results of unknown clinical significance, we have compared a selection of commercial test kits and have analysed their diagnostic value by data mining. Here we describe applications of data mining for the diagnosis of inflammatory and thrombotic induced acute central nervous processes and identification of various prognostic groups of cancer patients. Evaluation of laboratory results by data mining revealed a restricted suitability of chosen test parameters to reply diagnostic questions. Thereby, unnecessarily performed test systems could be removed from the diagnostic panel. Furthermore, computer assisted classification in positive and negative results according to clinical findings could be implemented.

1 Introduction

In clinical diagnostics of immunologically caused diseases and in processes involving alterations of humoral and cellular immune parameters, diagnostic of immunoparameters is hampered by missing reference test systems, unknown diagnostic relevance of recently introduced test systems, complex character of immunological findings and considerable costs to perform such tests. The aim of laboratory diagnostic, nevertheless, has to be to give clear diagnostic hints for clinical work. Therefore, we were interested to find a method that can efficiently evaluate the reliability of such tests. The focus of our work was to evaluate indications for immunological testing and to develop algorithms for machine based data interpretation. The method that supports requirements should fulfill the following criteria:

1. The method should be easy to use in laboratory work.
2. The resulting model should have explanation capability.
3. The methods should allow to select from the full set of parameters a subset of necessary diagnostic parameters.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 12–22, 2001.

4. The method should give us based on sound mathematical methods a quality measure (e.g. the error rate) for the learnt model.

Based on these requirements, we decided to use decision tree induction [1] since the method had shown valuable results in different medical diagnostic tasks. For our experiments we used the tool DECISION MASTER. The tool DECISION MASTER realizes different decision tree induction methods. It allows to evaluate the learnt decision tree by cross validation and it has a nice user interface. Here we have applied this method to serological and cellular data to investigate clinical significance.

2 Clinical Reliability of Antiphospholipide Autoantibodies (APA) in Diagnosis of Acute Cerebrovascular Processes

Anti-phospholipide antibodies (APA) cause recurrent venous and arterial events [2]. For the detection of these antibodies, a variety of immunoassays based on cardiolipin, the phospholipide cofactor β2-glycoprotein I (β2GPI), and phosphatidylserine are available [3]. Based on the fact, that detection of anti cardiolipin antibodies (ACA) is dependent on the presence of β2GPI [4], these antibodies against this antigen were expected to improve laboratory diagnostic in acute cardiovascular diseases [5].

Most assay systems are based on the ELISA principle (enzyme linked immuno sorbent assay), and are considered specific. Shortly, the corresponding antigen cardiolipin, β2GPI, or phosphatidylserine is bound to a microwell surface. After binding so the antigen, reactive antibodies in human serum dilutions can be detected by an enzyme conjugated secondary antibody that produces a colored product.

Nevertheless, assays of different manufacturers have been shown to yield highly different results. Therefore, we have compared a selection of commercial immunoassays for the detection of antibodies against cardiolipin, β2-glycoprotein I, and phosphatidylserine. We performed a concordance analysis and subsequently calculated a dendrogram (hierarchical tree plot) based on a complete linkage analysis. Nevertheless, comparison between different test systems does not provide information about clinical significance of yielded data. This is especially true in the case of APA-mediated acute central nervous syndroms [6]. To investigate validity of parameters for the diagnosis of processes underlying acute cerebrovascular diseases, we decided to use the data mining tool DECISION MASTER. We have applied this method here to ELISA data to investigate clinical significance.

2.1 Data Set

87 patients with inflammatory or thrombotic induced acute central nervous processes were characterized by the following methods:

1. clinical examination,
2. laboratory investigation,
3. magnetic resonance tomography,

4. computed tomography,
5. positron emission tomography, and
6. electrophysiological examinations.

By this way, an exact clinical diagnosis of the pathogenesis could be found. On the other hand, a quick, cheap and simplified method like an ELISA system would offer considerable advantages for early classification of patients and therapeutic decisions. Therefore, APA levels were determined by commercially provided ELISA systems based on plates coated with cardiolipin (n = 9 IgM and IgG, n = 5 IgA), β2GPI (n = 6 IgG, n = 5 IgM, n = 3 IgA), and phosphatidylserine (n = 1). Results were reported as absolute values and subsequently classified as negative and positive for each kit by test-specific cut-off values as provided by manufacturers. By this method, a data matrix was formed displaying positive (1) or negative (0) classifications for the same parameter as generated by the various assays. A sample of such a matrix is shown in Table. 1.

Table 1. Data matrix as generated by detecting IgG autoantibodies against cardiolipin by nine different test systems. Dependent on raw data and test-specific cut-off values as provided by manufacturers, results were classified as negative (0) or positive (1) ones. Obviously, highly divergent data were found.

sample	manufacturer 1	manufacturer 2	manufacturer 3	manufacturer 4	manufacturer 5	manufacturer 6	manufacturer 7	manufacturer 8	manufacturer 9
1	0	1	1	1	0	0	1	1	0
2	1	0	1	0	0	0	1	1	0
3	1	0	1	0	0	1	1	1	1
4	1	1	1	1	0	0	1	1	1
5	1	0	1	1	0	0	0	1	0
6	1	1	1	0	0	0	1	1	1
7	1	1	1	1	0	1	1	1	1
8	1	0	1	0	0	0	0	1	1
9	1	0	1	1	0	0	1	1	1
10	1	1	1	1	0	0	1	1	0
11	1	1	1	1	0	1	1	1	1
12	1	0	1	1	0	0	1	1	1
13	1	1	1	1	0	1	1	1	1
14	1	0	0	0	0	0	0	1	0
15	1	0	1	1	0	0	1	0	0
16	0	0	0	0	0	0	0	0	0
N	...	...	...	...	...	...	...	...	...

2.2 Data Analysis

First, data set was investigated for consistancy and missing values. The correlation between the different assays was assessed by correlation analysis (Spearman) and cut-off dependent classification (SPSS; Chicago, IL, U.S.A.) based on manufacturer's

data. Furthermore, dendrograms were calculated to determine linkage between test systems (Statistica; Statsoft, Tulsa, OK, U.S.A.).

To investigate the reliability of classification by autoantibody testing, patients were classified by clinical data into three diagnostic groups: arterial thrombosis [n = 65], veneous thrombosis [n = 6], and inflammatory process [n = 16]. Consequently, data were analyzed by the data mining tool DECISION MASTER [1]. Parameter were selected by information content. Discretization was performed by dynamic cut-point procedure. Reduced data tree was generated to minimal failure, and data evaluation was done by cross validation leaving one out.

2.3 Results

The determination of all APA levels revealed significant discrepancies. The estimated concordance between anti-cardiolipin-antibodies (ACA) was between only 4 and 66 %. Similarily, anti-β2GPI-assays were found to express a concordance between 12 and 70 % (Fig. 1).

Generation of dendrogram indicated several linked groups of test kits for each parameter. Analysis was performed for single linkage and Euclidean distances (Fig. 2).

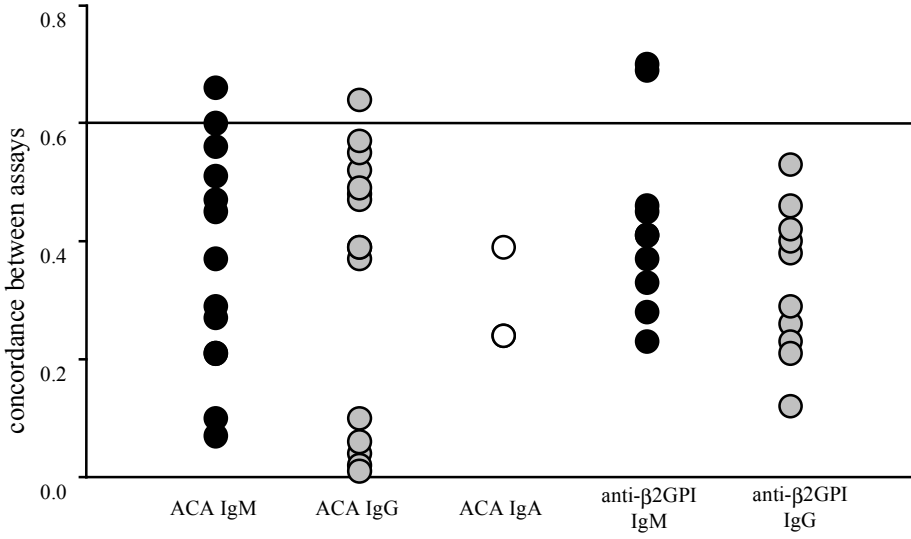


Fig. 1. Concordance between several commercial test systems for the investigation of antiphospholipide antibodies in sera of patients was mostly below 0.6; in fact most Elisa systems provided controverse data for more than 50 % of our samples.

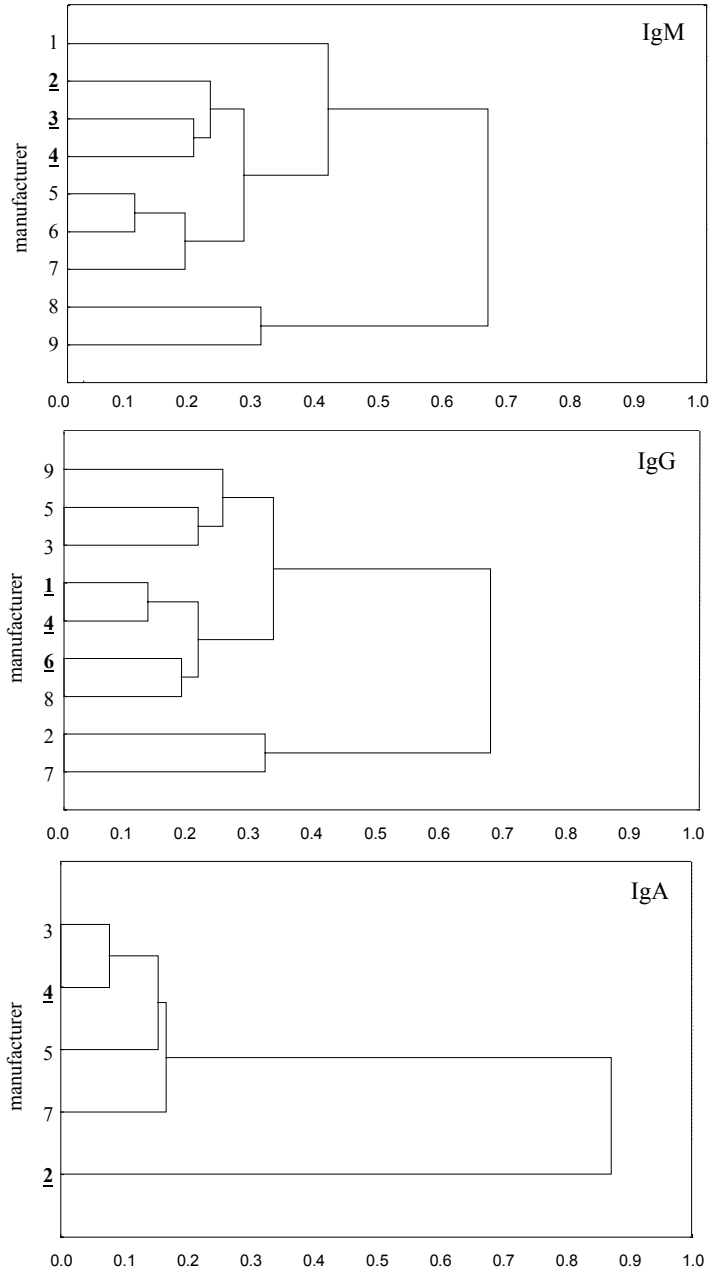


Fig. 2. Dendrogram presenting relationship as shown by distance (relative disagreement) between commercial test kits provided by different manufacturers for the detection of ACA (immunoglobulins IgM, IgG, IgA, respectively). Please note, that most closely linked assays in this figure do not correspond with the test systems providing best clinical information (indicated by bold underlined numbers).

Analysis of anti-β2GPI-assays revieled a comparable result. Beside singular closely related assays, most ELISA were only linked with a distance of 30 % (i.e. 0.3) or worse. Furthermore, we analysed the percentage of samples, classified as positive or negative findings based on manufacturers information. This analysis revealed similarly inconsistent data (Fig. 3).

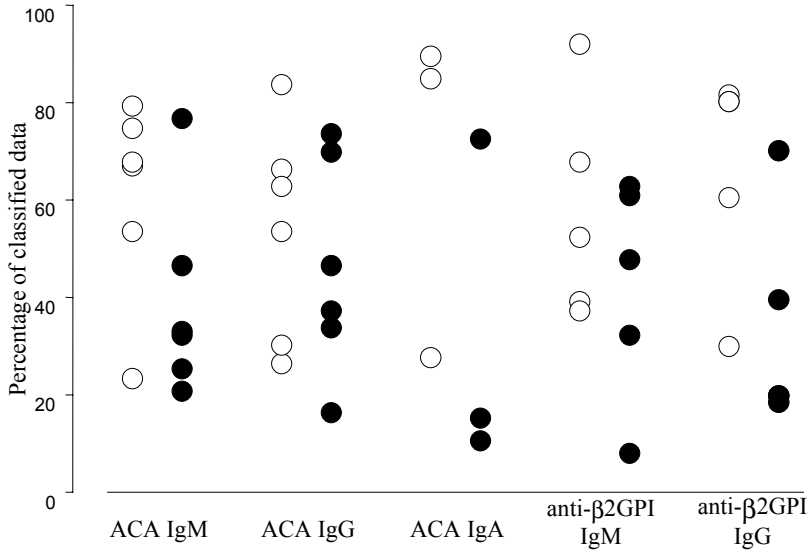


Fig. 3. Separation of serological values into negative (white) and positive (black) values according to the manufacturers data revealed overwhelming degree of divergent classification of sera. Minimal and maximal frequency of positive classified samples are shown by black and white bars, respectively. Assays for the detection of autoantibodies against cardiolipin (ACA) and β2-glycoprotein I (β2GPI) classified between 20 % and 80 % of matching samples as positive.

Because these data were extremely controversially, no singular immunoassay could be chosen as standard for evaluation. On the other hand, laboratory data should reflect clinical conditions. Therefore, clinical parameters and diagnosis of patients were taken to classify data. By data mining, data were investigated on clinical suitability.
By the data mining tool DECISION MASTER, decision trees for arterial thrombosis, veneous thrombosis, and inflammation were generated and evaluated. In Fig. 4, examples of unpruned decision trees as generated by analysis of an ACA-ELISA system for IgG and IgM antibodies are depicted. Although venous thrombosis could be determined clearly, arterial thrombosis and inflammation were not classified clearly. This is also true in the other systems investigated (Fig. 5). Decision trees revealed, that misclassifications were highly dependent on singular test systems and on clinical classification. Arterial thrombosis could not be identified with any of the ELISA's (probability of misclassification 43 to 60 %, except one β2GPI test at a probability of misclassification 30 %). Venous thrombosis could be detected clearly

with all systems investigated (no errors). Inflammations were recognized with a probability for misclassification of 18 to 45 % with cardiolipin, 13 % to 45 % with β2GPI, and 18 % with phosphatidylserine, respectively.

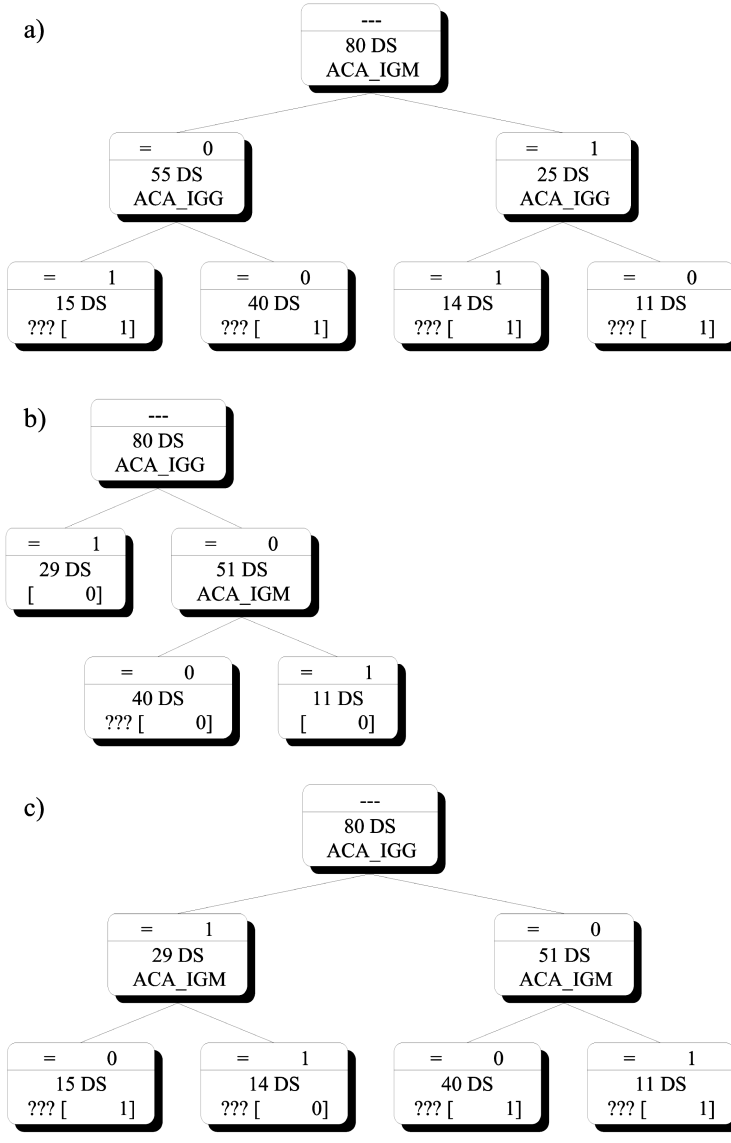


Fig. 4. Decision trees for the classification of patients suffering from arterial thrombosis [a)], veneous thrombosis [b)], and inflammation [c)] by ELISA's for the detection of ACA (IgM and IgG) as provided by one manufacturer. Evaluation of decision trees by cross validation leaving one out revealed an error rate of 41 %, 0 %, and 31 %, respectively.

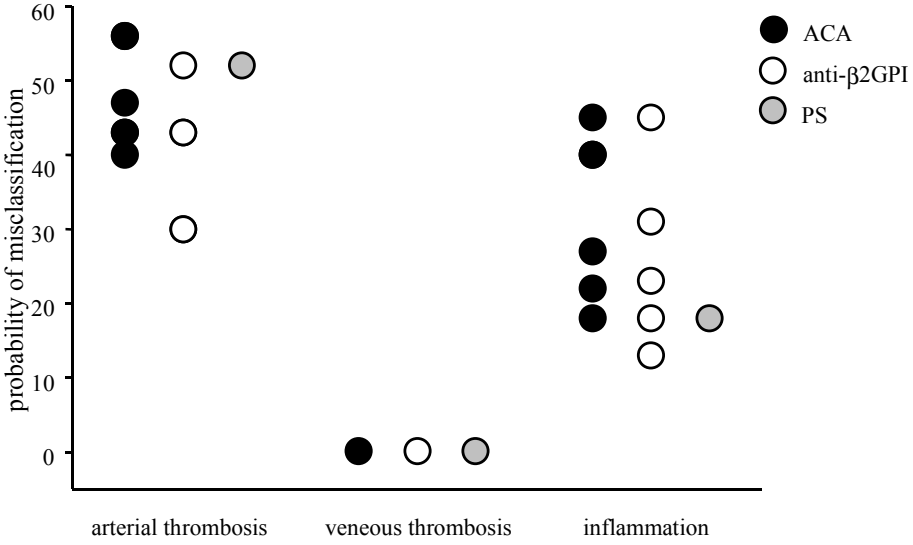


Fig. 5. Classification of arterial thrombosis, venous thrombosis as well as inflammation by using ELISA systems against several autoantigens. Simultaneous detection of IgG and IgM antibodies was essential (ACA: anticardiolipin antibodies; β2GPI: β2-glycoprotein I; PS: phosphatidylserine).

2.4 Conclusion

These data indicate that the determination of anti-cardiolipin and anti-β2GPI antibodies depends on the quality of the commercial kits used. Furthermore, the diagnostic efficiency of each commercial assay should be investigated. We conclude that commercially provided test systems differ substantially in their reliability to reflect clinical conditions. Furthermore, data mining has been shown to be a valuable tool for validation of test systems. Test systems differ substantially in their reliability to reflect clinical conditions.

3 Investigation of Immunophenotyping to Identify Prognosis of Patients Suspicious on Relapsing Cancer

Lymphocyte subpopulations reflect the health state of investigated subjects as well as a variety of diseases. Especially in the case of tumor care, immune system is considered to be crucial for clinical prognosis of patients and the risk of lethal outcome. Analysis of cellular immune parameters yields a large number of parameters, including percentages of different cell populations and their functional parameters [7].

To investigate validity of these parameters for the diagnosis of patients suspicious on relapsing carcinoma, we classified patients by survival without recidive for up to 5 years and by TNM grading of primary tumor.

3.1 Data Set

99 patients with tumors were included in the study (12 rectum carcinoma, 11 colon carcinoma, 2 gastric cancer, 65 mamma carcinoma). 90 of these tumors were classified as primary tumors, 9 as secondary. Treatment was performed as curative (n = 96) or palliative (n = 3) setting. All patients were scored according to the TNM system [8].

After a post-operative period of 2 to 5 months, blood samples were taken and flow cytometric analysis of lymphocyte subpopulations was performed to investigate T-, B-, NK- cells, furthermore T-helper- as well as T-cytotoxic cells, and activated T cells as shown by HLA-DR expression. Flow cytofluorometric analysis was performed to investigate lymphocyte subpopulations.

Cells were stained with fluorescein isothiocyanate or phycoerythrin (FITC or PE)-conjugated murine monoclonal antybodies (mAbs) directed against human cell surface markers. The following mAbs were used for two-color analysis: anti-CD25-PE (Beckman-Coulter, Krefeld, Germany), anti-CD3-FITC, anti-CD4-PE, anti-CD8-PE, anti-CD14-PE, anti-CD16-PE, anti-CD19-PE, anti-CD45-FITC, anti-CD56-PE, anti-HLA-DR-PE (BD Biosciences, Heidelberg, Germany), and anti-IgG1-FITC/anti-IgG2a-PE (BD Biosciences) as isotype control. The samples were analyzed on a FACScan® (BD Biosciences) instrument based on their size, granularity, and specific two-colour fluorescence describing cellular lineage. By this way, absolute number and relative frequency of different lymphocyte subpopulations in peripheral blood can be calculated.

3.2 Data Analysis

First, data set was investigated for consistency and missing values. To identify statistical differences between groups of patients, Mann-Whitney U-test was performed. Subsequently, data were analyzed by the data mining tool DECISION MASTER [1]. Parameter were selected by information content. Discretization was performed by dynamic cut-point procedure. Reduced data tree was generated to minimal failure, and data evaluation was done by cross validation leaving one out.

3.3 Results

By statistical analysis, no statistical differences between groups of patients could be identified. Mann-Whitney U-test did not reveal any significant differences. By analysis with the data mining tool DECISION MASTER [1], none of the laboratory values could be shown to allow the calculation of one of the classification criteria.

3.4 Conclusion

Although the immune system, especially lymphatic cells, are closely connected with tumor defense of the body, no differentiation between different groups of patients or

prognostic data could be found for the selected parameters. Probably, another data more closely connected to functional parameters of tumor defense should be determined and included in such an analysis.

4 Summary

Data mining has shown previously to provide a valuable tool to investigate clinical data for diagnostic knowledge [9]. This can be applied to the processing of images [10-12] as well as to the investigation of laboratory results [1]. Here we present the application of data mining to the analysis of laboratory results of a clinical-immunological laboratory. By means of data mining, extraction of knowledge out of the data was possible in one of the investigated cases, allowing the generation of a dicision tree. This enables us to select the most suitable test system as well as to generate clear-text recommendations to the clinical doctor.

In a second setting, no relevant data were found inside a data collection. On the one hand, this fits well the characteristics of the provided laboratory findings characterized by missing differences between groups of patients. On the other hand, this underlines the fact, that data mining only extracts relevant data, and must give a negative result in an unappropriate selection of data.

Acknowledgements

The authors would like to thank Dr. Petra Perner for valuable help in analysing data, and Dr. Jens Förster as well as Dr. Steffen Leinung for providing clinical information according the investigated patients. Authors are grateful to Dr. Thomas Keller for creating the hierarchical tree plot.

References

1. Data mining tool Decision Master® http://www.ibai-solutions.de
2. Hughes, G.R. The anticardiolipin syndrome: Clin. Exp. Rheumatol. 3 (1985), 285-286.
3. Matsuura, E. Assay principles of antiphospholipid antibodies and heterogeneity of the antibodies: Rinsho Byori 48 (2000), 317-322.
4. Ichikawa, K., Khamashta, M.A., Koike, T., Matsuura, E., Hughes, G.R. beta 2-Glycoprotein I reactivity of monoclonal anticardiolipin antibodies from patients with the antiphospholipid syndrome: Arthritis Rheum. 37 (1994), 1453-1461.
5. Gharavi, A.E., Sammaritano, L.R., Wen, J., Elkon, K.B. Induction of antiphospholipid autoantibodies by immunization with beta 2 glycoprotein I (apolipoprotein H): J. Clin. Invest. 90 (1992), 1105-1109.
6. Wöhrle, R., Matthias, T., von Landenberg, P., Oppermann, M., Helmke, K., Förger, F. Comparing different anti-cardiolipin- and anti-β2-glycoprotein-I-antibody-ELISA in autoimmune diseases, in: Conrad, K., Humbel, R.-L., Meurer,

M., Shoenfeld, Y., Tan E. M. (eds.) Autoantigens and Autoantibodies: Diagnostic tools and clues to understanding autoimmunity. Lengerich, Berlin, Riga, Rom, Wien, Zagreb: Pabst Science Publishers (2000), 410-411.
7. Sack, U., Rothe, G., Barlage, S., Gruber, R., Kabelitz, D., Kleine, T. O., Lun, A., Renz, H., Ruf, A., Schmitz, G. Durchflusszytometrie in der Klinischen Diagnostik. J. Lab. Med. 24 (2000), 277-297.
8. Appere de Vecchi, C., Brechot, J.M., Lebeau, B. The TNM classification. Critical review. Rev. Mal. Respir. 15 (1998), 323-332.
9. Perner, P., Trautzsch, S. Wissenakquisition in der medizinischen Diagnose mittels Induktion von Entscheidungsbäumen, Zeitschrift Künstliche Intelligenz, 3 (1997), 32-33.
10. Perner, P. Mining Knowledge in Medical Image Databases, in: Data Mining and Knowledge Discovery: Theory, Tools, and Technology, Belur V. Dasarathy (eds.), Proceedings of SPIE 4057 (2000), 359-369.
11. Perner, P. An architecture for a CBR image segmentation system. Engineering Applications of Artificial Intelligence 12 (1999), 749-759.
12. Perner, P. Image analysis and classification of HEp-2 cells in fluorescent images. Proceedings of the 14th International Conference on Pattern Recognition, Brisbane Australiy, IEEE Computer Society Press Vol. II (1998), 1677-1679.

Temporal Abstractions and Case-Based Reasoning for Medical Course Data: Two Prognostic Applications

Rainer Schmidt, Lothar Gierl

Institut für Medizinische Informatik und Biometrie,
Universität Rostock, D-18055 Rostock, Germany

Abstract. We have developed a method for analysis and prognosis of multiparametric kidney function courses. The method combines two abstraction steps (state abstraction and temporal abstraction) with Case-based Reasoning. Recently we have started to apply the same method in the domain of Geomedicine, namely for the prognosis of the temporal spread of diseases, mainly of influenza, where just one of the two abstraction steps is necessary, that is the temporal one. In this paper, we present the application of our method in the kidney function domain, show how we are going to apply the same ideas for the prognosis of the spread of diseases, and summarise the main principles of the method.

1 Introduction

At our ICU, physicians daily get a printed renal report from the monitoring system NIMON [1] which consists of 13 measured and 33 calculated parameters of those patients where renal function monitoring was applied. The interpretation of all reported parameters is quite complex and special knowledge of the renal physiology is required.

So, the aim of our knowledge-based system ICONS is to give an automatic interpretation of the renal state and to elicit impairments of the kidney function on time. That means, we need a time course analysis of many parameters without any well-defined standards. Although much research has been performed in the field of conventional temporal course analysis in the recent years, none of them is suitable for this problem. Allen‘s theory of time and action [2] is not appropriate for multiparametric course analysis, because time is represented as just another parameter in the relevant predicates and therefore does not give necessary explicit status [3]. Traditional time series techniques [4] with known periodicity work well unless abrupt changes, but they do not fit in a domain characterised by possibilities of abrupt changes and a lack of well-known periodicity. An ability of RÉSUMÉ [5] is the abstraction of many parameters into one single parameter and to analyse the course of this abstracted parameter. However, the interpretation of a course requires complete domain knowledge. Haimowitz and Kohane [6] compare many parameters of current courses with well-known standards.

However, in the kidney function domain, neither a prototypical approach in ICU settings is known nor exists complete knowledge about the kidney function.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 23–34, 2001.

Especially, knowledge about the behaviour of the various parameters over time is yet incomplete. So we had to design our own method to deal with course analysis of multiple parameters without prototypical courses and without a complete domain theory.

Our method combines abstracting many parameters into one general parameter and subsequently analysing the course of this parameter with the idea of comparing courses. However, we do not compare with well-known standards (because they are not yet known), but with former similar courses.

At present we have started to apply the same method in a completely different domain, for the prognosis of the spread of diseases. Since in the geomedical domain we do not have daily multiple parameter sets, but just one single parameter per week (namely incidences of a disease), the first abstraction step is left out. However, the main idea remains the same, namely to describe temporal courses by different trends and to use the parameters of these trend descriptions to retrieve former similar cases from a case base.

2 Prognosis of Kidney Function Courses

Our method to analyse and forecast kidney function courses is shown in Fig.1. First, the monitoring system NIMON gets 13 measured parameters from the clinical chemistry and calculates 33 meaningful kidney function parameters. Since it was impossible to make complex relations among all parameters visible, we decided to abstract these parameters. For this data abstraction we use states of the renal function, which determine states of increasing severity beginning with a normal renal function and ending with a renal failure. Based on these state definitions, we determine the appropriate state of the kidney function per day. We use transitions of these states of one day to the state of the respectively next day to generate three different trend descriptions. Subsequently, we use Case-based Reasoning retrieval methods [7, 8, 9, 10] to search for similar courses. Together with the current course we present the most similar courses as comparisons to the user, their course continuations serve as prognoses.

Since ICONS offers only diagnostic and prognostic support, the user has to decide about the relevance of all displayed information. When presenting a comparison of a current course with a similar one, ICONS supplies the user with the ability to access additional renal syndromes, which sometimes describe supplementary aspects of the kidney function, and the courses of single parameter values during the relevant time period.

2.1 State Abstraction

Since the 46 parameter values provided by the monitoring system NIMON are based on just 13 measured data, is was rather easy for domain experts to define kidney function states by about a dozen parameters. These states are characterised by not exactly, but nearly the same parameters. Since creatinin clearence is the leading kidney function parameter, the kidney function states are defined by one obligatory

(creatinin clearence) and between 9 and 12 optional conditions for the selected renal parameters. The conditions are either intervals or greater respectively smaller relations. For those states that satisfy the obligatory condition we calculate a similarity value concerning the optional conditions. We use a variation of Tversky's [7] measure of dissimilarity between concepts. Only if two or more states are very probable, which means that their dissimilarity difference is very small, ICONS presents the states under consideration to the user. These states are sorted according to their computed similarity values and they are presented together with information about the satisfied and not satisfied optional conditions. The user has to decide which of them fits best. For detailed information about this step including evaluation results see [11].

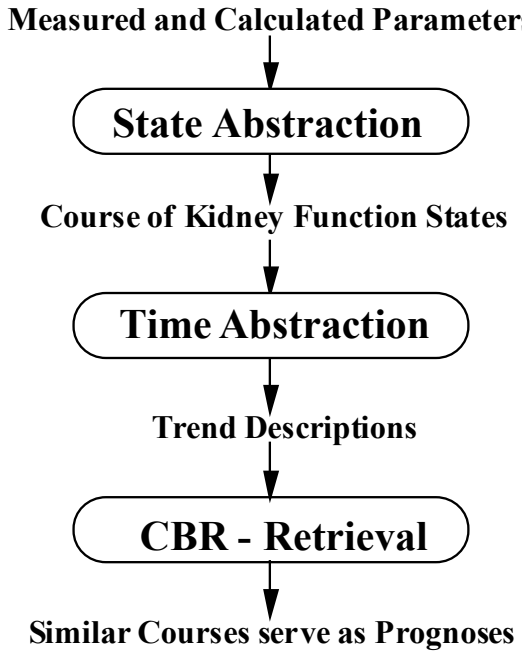


Fig. 1. The prognostic model for ICONS

2.2 Temporal Abstraction

First, we defined five assessments for the transition of the kidney function state of one day to the state of the respectively next day. These assessments are related to the grade of renal impairment:

steady: both states have the same severity value.

increasing: exactly one severity step in the direction towards a normal function.

sharply increasing: at least two severity steps towards a normal function.

decreasing: exactly one severity step in the direction towards a kidney failure.

sharply decreasing: at least two severity steps towards a kidney failure.

These assessment definitions are used to determine the state transitions from one qualitative value to another. Based on these state transitions, we generate three trend descriptions.

T1, short-term trend:=	current state transition
T2, medium-term trend:=	looks recursively back from the current state transition to the one before and unites them if they are both of the same direction or if one of them has a "steady" assessment
T3, long-term trend:=	characterises the whole considered course of at most seven days

For the long-term trend description, we additionally defined four new assessments. If none of the five assessments described above fits the complete considered course, we attempt to fit one of these four definitions in the following order:

alternating: at least two up and two down transitions and all local minima are equal.

oscillating: at least two up and two down transitions.

fluctuating: distance of the highest to the lowest severity state value is greater than 1.

nearly steady: the distance of the highest to the lowest severity state value equals one.

Why these four trend descriptions? There are domain specific reasons for defining the short-, medium- and long-term trend descriptions T1, T2 and T3. If physicians evaluate courses of the kidney function, they consider at most one week prior to the current date. Earlier renal function states are irrelevant for the current situation of a patient. Most relevant information is derived from the current function state, the current development and sometimes a current development within a slightly longer time period. That means, very long trends are of no interest in this domain.

The short-term trend description T1 expresses the current development. For longer time periods, we have defined the medium- and long-term trend descriptions T2 and T3, because there are two different phenomena to discover and for each, a specific technique is needed. T2 can be used for detecting a continuous trend independent of its length, because equal or steady state transitions are recursively united beginning with the current one. As the long-term trend description T3 describes a well-defined time period, it is especially useful for detecting fluctuating trends.

2.3 Retrieval

We use the parameters of the three trend descriptions and the current kidney function state to search for similar courses. As the aim is to develop an early warning system, we need a prognosis. For this reason and to avoid a sequential runtime search along the whole cases, for each day a patient spent on the intensive care unit we store a course of the previous seven days and a maximal projection of three days.

Since many different continuations are possible for the same previous course, it is necessary to search for similar courses and different projections. Therefore, we divided the search space into nine parts corresponding to the possible continuation directions. Each direction forms an own part of the search space. During the retrieval these parts are searched separately and each part may provide at most one similar case. The at most 9 similar cases of the 9 parts are presented in the order of their computed similarity values. Fig.2. shows such a presentation.

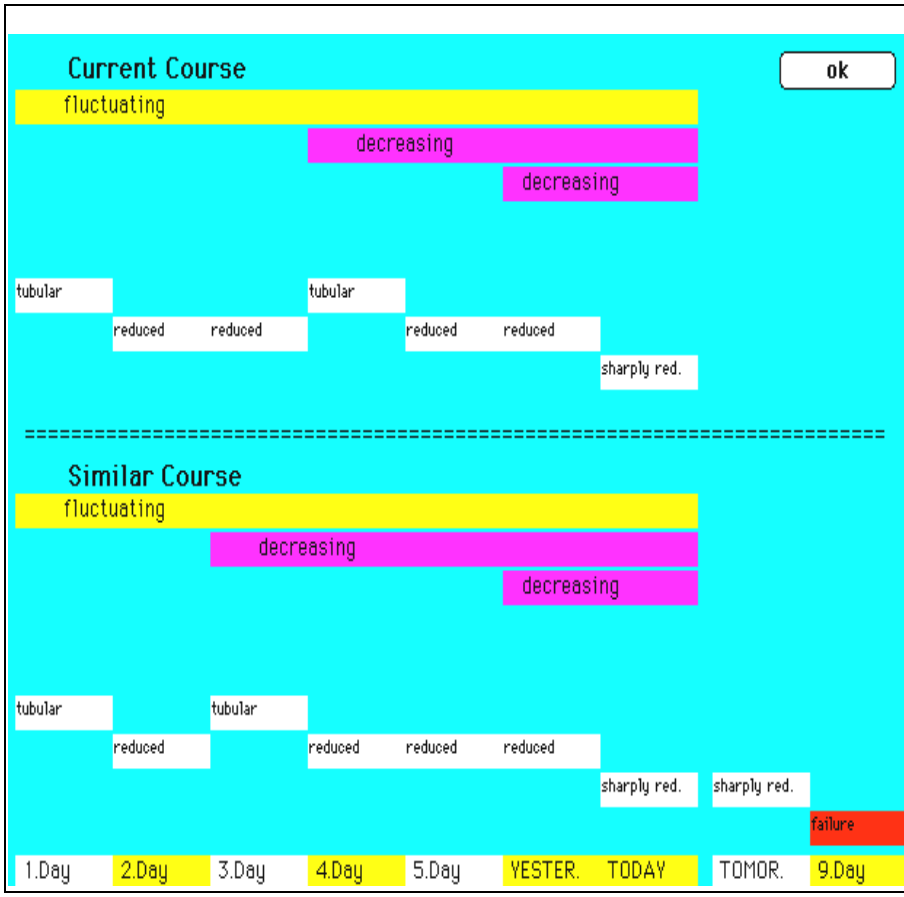


Fig.2. Comparative presentation of a current and a similar course. In the lower part of each course the (abbreviated) kidney function states are depicted. The upper part of each course shows the deduced trend descriptions.

For each part, the retrieval consists of two steps. First we search with an activation algorithm concerning qualitative features. Our algorithm differs from the common spreading activation algorithm [8] mainly due to the fact that we do not use a net for the similarity relations. Instead, we explicitly have defined activation values for each possible feature value. This is possible, because on this abstraction level there are only ten dimensions (see the left column of Table 1.) with at most six values.

Table 1. Retrieval dimensions and their activation values

Dimensions	Activation values
Current state	15, 7, 5, 2
Assessment T1	10, 5, 2
Assessment T2	4, 2, 1
Assessment T3	6, 5, 4, 3, 2, 1
Length T1	10, 5, 3, 1,
Length T2	3, 1
Length T3	2, 1
Start state T1	4, 2
Start state T2	4, 2
Start state T3	2, 1

The right column of Table 1. shows the possible activation values for the description parameters. E.g. there are four activation values for the current kidney function state: courses with the same current state as the current course get the activation value 15, those cases whose distance to the current state of the current course is one step in the severity hierarchy get 7 and so forth.

Subsequently, we check the list of cases, sorted according to their computed similarity, with a similarity criterion until one case fulfils it. This criterion looks for sufficient similarity, because even the most similar course may differ from the current one significantly [9]. This may happen at the beginning of the use of ICONS, when there are only a few cases known to ICONS, or when the current course is rather exceptional.

2.4 Learning

Prognosis of multiparametric courses of the kidney function for ICU patients is a domain without a medical theory. Moreover, we can not expect such a theory to be formulated in the near future. So we attempt to learn prototypical course pattern. Therefore, knowledge on this domain is stored as a tree of prototypes with three levels and a root node. Except for the root, where all not yet clustered courses are stored, every level corresponds to one of the trend descriptions T1, T2 or T3. As soon as enough courses that share another trend description are stored at a prototype, we create a new prototype with this trend. At a prototype at level 1, we cluster courses that share T1, at level 2, courses that share T1 and T2 and at level 3, courses that share all three trend descriptions. We can do this, because regarding their importance, the short-, medium- and long-term trend descriptions T1, T2 and T3 refer to hierarchically related time periods. T1 is more important than T2 and T3.

So, before the retrieval starts we search for a prototype that has most of the trend descriptions in common with the current course. The search begins at the root with a check for a prototype with the same short-term trend description T1. If such a prototype can be found, the search goes on below this prototype for a prototype that

has the same trend descriptions T1 and T2, and so on. The retrieval starts below the last accepted prototype. For details about the prototype architecture in ICONS see [11], and for details about the general role of prototypes for learning and for structuring case bases within medical knowledge-based systems see [12].

3 Prognosis of the Spread of Diseases

Recently we have started the TeCoMed project. The aim of this project is to discover regional health risks in the German federal state Mecklenburg-Western Pomerania. Furthermore, the current situation of infectious diseases should be presented on the internet. So, we have begun to develop a program to analyse and forecast the temporal spread of infectious diseases, especially of influenza (Fig.3. shows the temporal spread of influenza in Germany). As a method we use Case-based Reasoning again to search for former, similar developments.

However, there are some differences in comparison to the kidney function domain:

Here, a state abstraction is unnecessary and impossible, because now we have got just one parameter, namely weekly incidences of a disease. So, we have to deal with courses of integer values instead of nominal states related to a hierarchy. And the data are not measured daily, but weekly. Since we believe that courses should reflect the development of four weeks, courses consist of 4 integer values.

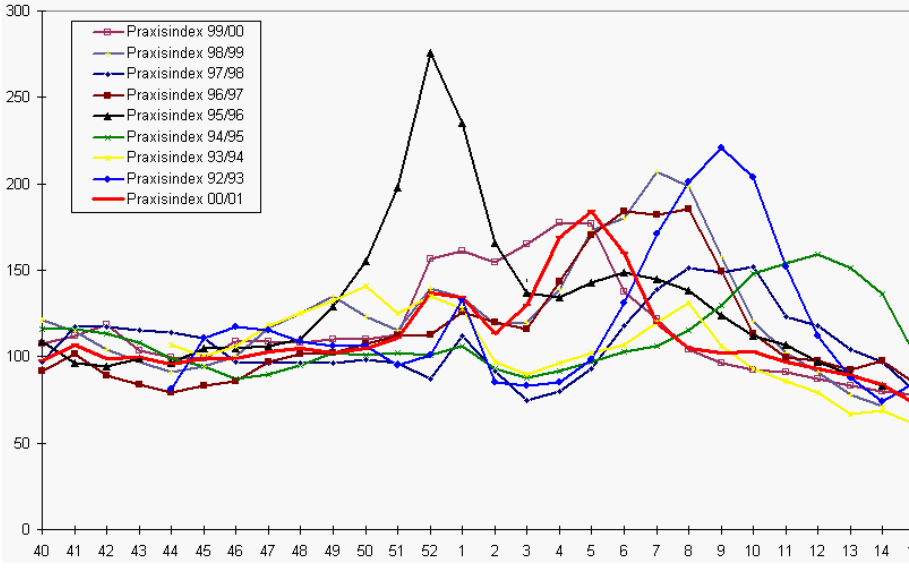


Fig.3. Temporal spread of influenza during the last ten influenza periods in Germany, depicted on the web by the influenza working group (http://www.dgk.de/agi/). Horizontally the weeks are depicted, vertically a "Praxisindex", which means the number of influenza patients per thousand people visiting a doctor.

So, our prognostic model for TeCoMed (Fig. 4) is slightly different than the one for ICONS. Again, we use three trend descriptions. They are the assessments of the developments from last week to this week (T1), from last but one week to this week (T2) and so forth. For retrieval, we use these three assessments (nominal valued) plus the four weekly data (integers). We use these two sorts of parameters, because we intend to ensure that a current query course and an appropriate similar course are on the same level (similar weekly data) and that they have similar changes on time (similar assessments).

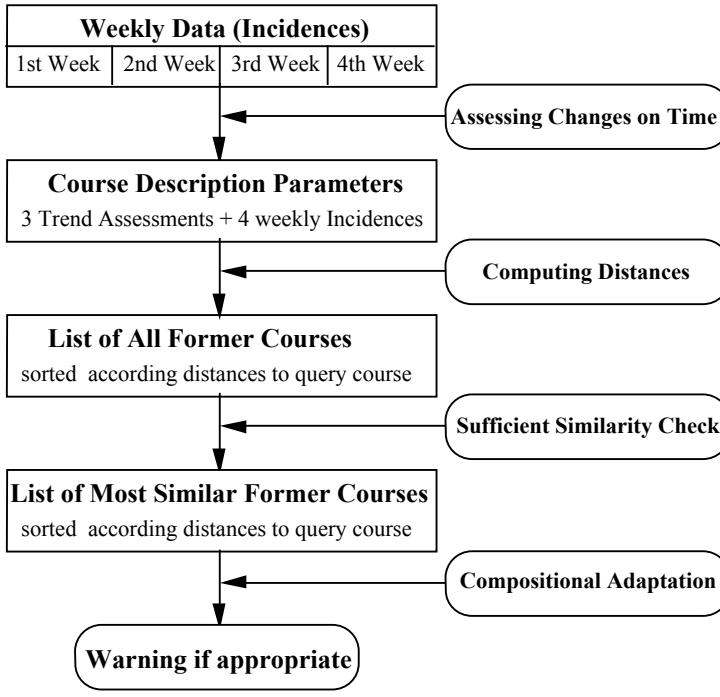


Fig.4. The prognostic model for TeCoMed

3.1 Searching for Similar Courses

So far, we sequentially compute distances between a query course and all courses in the case base. The considered attributes are the trend assessments and the weekly incidences. For each trend we have defined separate assessments based on the percentage of the changes of the weekly data. For example, we assess the third trend T3 as "threatening increase" if the data of the current week is at least 50% higher than the data three weeks ago. When comparing a query course with a former one, equal assessments are valued as 1.0 and neighbouring ones as 0.5. Again the current trend (T1) is weighted higher than longer ones (T2, T3).

For the weekly data, we compute differences between the incidences of a query and a former course and weight them with the number of the week within the four weeks course (e.g. the first week gets the weight 1.0, the current week gets 4.0).

To bring both sorts of parameters together on equal terms we multiply the computed assessment similarity with the doubled mean value of the weekly data. The result of this similarity computation is a list of all 4-weeks courses in the case base sorted according to their distances with respect to the query course.

As we have done in the kidney function domain, we reduce this list by checking for sufficient similarity [13]. For the sum of the three assessments we use a distance threshold which should not be overstepped. Concerning the four weekly incidences we have defined individual constraints that allow specific percentage deviations from the query case data. At present we are attempting to learn good settings for the parameters of these similarity constraints by using former courses in retrospect.

3.2 Adaptation

For adaptation, we apply compositional adaptation [14], because now our goal is not to present the most similar courses to the user again, but to send warnings - when appropriate – e.g. against a forthcoming influenza wave to interested people (practitioners, pharmacists etc.). We marked the moments of the former 1-year courses where in retrospect warnings would have been appropriate. So, the reduced list of similar 4-weeks courses can be split in two lists, namely concerning the question if a warning would have been appropriate or not. For both of these new lists we compute the sum of the reciprocal distances of their courses in respect to the query course. Subsequently, the decision about the appropriateness to give warnings depends on the question, which of the two sums is bigger.

However, so far the definitions for retrieval and for adaptation are just based on few experiments and have to be funded or modified if necessary by further experiments and experiences. For adaptation, further information like the spatial spread of diseases and the local vaccination situation should be considered in the future. For retrieval, we again intend to structure the case base by generalising from single courses to prototypical ones. In ICONS we could do this by exactly matching nominal description parameters. However, a method to do this when the parameters are mainly integers as in TeCoMed still has to be generated.

4 Generalisation of Our Prognostic Method

Aamodt and Plaza have developed a well-known Case-based Reasoning cycle, which consists of four steps: retrieving former similar cases, adapting their solutions to the current problem, revising a proposed solution, and retaining new learned cases [15]. Fig.5. shows an adaptation of this cycle to our medical temporal abstraction method.

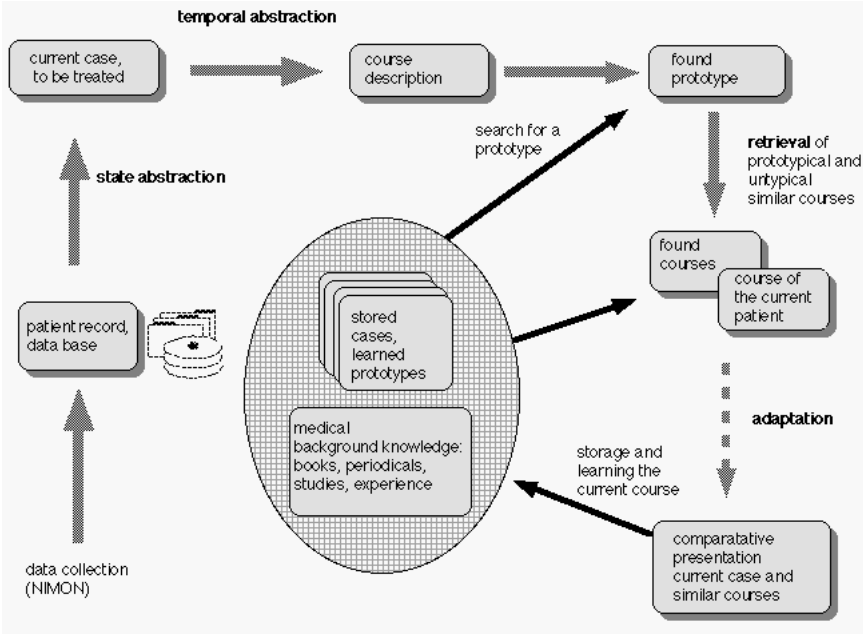


Fig.5. The Case-based Reasoning cycle adapted to medical temporal abstraction

Since the idea of both of our applications is to give information about a specific development and its probable continuation, we do not generate a solution that should be revised by a user. So, in comparison to the original CBR cycle our method does not contain a revision step.

In our applications the adaptation just consists of a sufficient similarity criterion. However, in the TeCoMed project we intend to broaden the adaptation to further criteria and information sources.

On the other hand, we have added some steps to the original CBR cycle. For multiple parameters (as in ICONS) we propose a state abstraction. For a single parameter (as in TeCoMed) this step should be dropped. The next step, a temporal abstraction, should provide some trend descriptions, which should not only help to analyse current courses, but the description parameters should also be used for retrieving similar courses. A domain dependent similarity has to be defined for retrieval and for the sufficient similarity criterion, which can be viewed as part of the adaptation step.

We believe that - at least in the medical domain - prototypes are an appropriate knowledge representation form to generalise from single cases [16]. They help to structure the case base, to guide and to speed up the retrieval, and to get rid of redundant cases [12]. Especially for course prognoses they are very useful, because otherwise too many very similar, former courses would remain in the case base. So, the search of the fitting prototype becomes a sort of preliminary selection, before the main retrieval takes only those cases into account that belong to the determined prototype.

5 Conclusion

In this paper we have proposed a prognostic method for temporal courses, which combines temporal abstractions with Case-based Reasoning. We have presented the prognostic model for prognosis of kidney function courses in ICONS and we have presented first steps of applying the same method for the prognosis of the spread of diseases in the TeCoMed project. Though there are some differences between both applications the main principles are the same. Temporal courses can be characterised by domain dependent trend descriptions. The parameters of these descriptions are used to determine the similarities of a current query course to former courses. Subsequently, we check the retrieved former courses with a criterion that guarantees sufficient similarity. So, in both applications we come up with a list of the most similar cases. In ICONS we present them to the user, while in TeCoMed, we apply compositional adaptation to decide whether early warnings are appropriate or not.

Based on the experiences with these two applications, we have proposed a prognostic method for temporal courses, which combines temporal abstraction with Case-based Reasoning. Furthermore, we have adapted the well-known Case-based Reasoning cycle of Aamodt and Plaza to temporal course prognosis.

References

1. Wenkebach,U., Pollwein, B., Finsterer, U.: Visualization of large datasets in intensive care. Proc Annu Symp Comput Appl Med Care (1992) 18-22
2. Allen, J.P.: Towards a general theory of action and time. Artificial Intelligence 23 (1984) 123-154
3. Keravnou, E.T.: Modelling Medical Concepts as Time Objects. In: Barahona, P., Stefanelli, M., Wyatt, J. (eds.): Artificial Intelligence in Medicine, Lecture Notes in Artificial Intelligence, Vol. 934, Springer-Verlag, Berlin Heidelberg New York (1995) 67-78
4. Robeson, S.M., Steyn, D.G.: Evaluation and comparison of statistical forecast models for daily maximum ozone concentrations. Atmospheric Environment 24 B 2 (1990) 303-12
5. Shahar, Y., Musen, M.A.: RÉSUMÉ: A Temporal-Abstraction System for Patient Monitoring. Computers and Biomedical Research 26 (1993) 255-273
6. Haimowitz, I.J., Kohane, I.S.: Automated Trend Detection with Alternate Temporal Hypotheses. In: Bajcsy, R. (ed.): Proceedings of IJCAI-93, Morgan Kaufmann Publishers, San Mateo, CA (1993) 146-151
7. Tversky, A.: Features of Similarity. Psychological Review 84 (1977) 327-352
8. Anderson, J.R.: A theory of the origins of human knowledge. Artificial Intelligence 40, Special Volume on Machine Learning (1989) 313-351
9. Smyth, B., Keane, M.T.: Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. Artif Intelligence 102 (1998) 249-293
10. DeSarbo, W.S., Johnson, M.D., Manrei, A.K., Manrai, L.A., Edwards, E.A.: TSCALE: A new multidemensional scaling procedure based on Tversky's contrast model. Psychometrika57 (1992) 43-69
11. Schmidt, R., Pollwein, B., Gierl, L.: Medical multiparametric time course prognoses applied to kidney function assessments. Int J Medical Inform 53 (2-3) (1999) 253-264

12. Schmidt, R., Pollwein, B., Gierl, L.: Experiences with Case-Based Reasoning Methods and Prototypes for Medical Knowledge-Based Systems. In: Horn, W., Shahar, Y., Lindberg, G., Andreassen, S., Wyatt, J. (eds.): Artificial Intelligence in Medicine, AIMDM'99, Lecture Notes in Artificial Intelligence, Vol. 1620, Springer-Verlag, Berlin Heidelberg New York (1999) 124-132
13. Smyth, B., Keane, M.T.: Adaptation-Guided Retrieval: Questioning the Similarity Assumption in Reasoning. Artif Intelligence 102 (1998) 249-293
14. Wilke, W., Smyth, B., Cunningham, P.: Using Configuration Techniques for Adaptation In: Lenz, M., et al. (eds): Case-Based Reasoning Technology, Lecture Notes in Artificial Intelligence, Vol. 1400, Springer-Verlag, Berlin Heidelberg New York (1998) 139-168
15. Aamodt, A., Plaza, E.: Case-Based Reasoning: Foundation Issues. Methodological Variation- and System Approaches. AICOM 7;1 (1994) 39-59
16. Schmidt, R., Gierl, L.: Case-based Reasoning for Medical Knowledge-based Systems. Medical Informatics Europe. In: Hasman, A., Blobel, B., Dudeck, J., Engelbrecht, R., Gell, G., Prokosch, H.-U. (eds.): Medical Infobahn for Europe, Proceedings of MIE2000 and GMDS2000, IOS Press, Amsterdam (2000) 720-725

Are Case-Based Reasoning and Dissimilarity-Based Classification Two Sides of the Same Coin?

Petra Perner

Institute of Computer Vision and applied Computer Sciences
Arno-Nitzsche-Str. 45, 04277 Leipzig
ibaiperner@aol.com, http://www.ibai-research.de

Abstract. Case-Based Reasoning is used when generalized knowledge is lacking. The method works on a set of cases formerly processed and stored in the case base. A new case is interpreted based on its similarity to cases in the case base. The closest case with its associated result is selected and presented as output of the system. Recently, Dissimilarity-based Classification has been introduced due to the curse of dimensionality of feature spaces and the problem arising when trying to make image features explicitly. The approach classifies samples based on their dissimilarity value to all training samples. In this paper, we are reviewing the basic properties of these two approaches. We show the similarity of Dissimilarity based Classification to Case-Based Reasoning. Finally, we conclude that Dissimilarity based Classification is a variant of Case-Based Reasoning and that most of the open problems in Dissimilarity-based Classification are research topics of Case-Based Reasoning.

1 Introduction

Case-Based Reasoning (CBR) has been developed within the artificial intelligence community. It uses past experiences to solve new problems. Therefore, past problems are stored as cases in a case base and a new case is classified by determining the most similar case from the case base. Although, CBR has been used with great success, for image related applications the examples are rare [1]-[7] and not well known within the pattern recognition community.

Recently, Dissimilarity-based classification (DSC)[8][9] has been introduced within the pattern recognition community. Objects are represented by their dissimilarity value to all objects in the case base. Classification is done based on the dissimilarity values. It is argued that dissimilarity based representations of objects are simpler to access than feature based representations and that this approach helps to comeover the curse of dimensionality of feature spaces.

In this paper, we are reviewing the basic properties of these two approaches. CBR is described in detail in Section 2. DSC is reviewed in Section 3. Finally, we compare these two approaches in Section 4. We show that DSC relies on the same basic idea as CBR. While CBR has covered all aspects of the development of a CBR system which range from fundamental theory to software engineering aspects, DSC work is very preliminary and does not cover all aspects that make such systems work in practice.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 35–51, 2001.

Finally, we can conclude that DSC is a special variant of CBR that is influenced by the traditional ideas of pattern recognition.

2 Case-Based Reasoning

Rule-based systems or decision trees are difficult to utilize in domains where generalized knowledge is lacking. However, often there is a need for a prediction system even though there is not enough generalized knowledge. Such a system should a) solve problems using the already stored knowledge and b) capture new knowledge making it immediately available to solve the next problem. To accomplish these tasks case based reasoning is useful. Case-based reasoning explicitly uses past cases from the domain expert's successful or failing experiences.
Therefore, case-based reasoning can be seen as a method for problem solving as well as a method to capture new experiences. It can be seen as a learning and knowledge discovery approach since it can capture from new experiences some general knowledge such as case classes, prototypes and some higher level concepts. The theory and motives behind CBR techniques are described in depth in [10][11][43]. An overview about recent CBR work can be found in [12].
To point out the differences between a CBR learning system and a symbolic learning system, which represents a learned concept explicitly, e.g. by formulas, rules or decision trees, we follow the notion of Wess et al. [13]: A case-based reasoning system describes a concept C implicitly by a pair *(CB, sim)*. The relationship between the case base *CB* and the measure *sim* used for classification may be characterized by the equation:

Concept = Case Base + Measure of Similarity

This equation indicates in analogy to arithmetic that it is possible to represent a given concept *C* in multiple ways, i.e. there exist many pairs $C= (CB_1, sim_1), (CB_2, sim_2), ..., (CB_i,sim_i)$ for the same concept *C*. Furthermore, the equation gives a hint how a case-based learner can improve its classification ability. There are three possibilities to improve a case-based system. The system can

- store new cases in the case base *CB,*
- change the measure of similarity *sim*,
- or change *CB* and *sim*.

During the learning phase a case-based system gets a sequence of cases $X_1, X_2, ..., X_i$ with $X_i= (x_i, class (x_i))$ and builds a sequence of pairs $(CB_1, sim_1), (CB_2, sim_2), ..., (CB_i, sim_i)$ with $CB_i \subseteq \{X_1, X_2, ..., X_i\}$. The aim is to get in the limit a pair (CB_n, sim_n) that needs no further change, i.e. $\exists n \ \forall m \geq n \ (CB_n, sim_n) = (CB_m, sim_m)$, because it is a correct classifier for the target concept *C*.

2.1 The Case-Based Reasoning Process

The CBR reasoning process is comprised of six phases (see Figure 1):

- Current problem description
- Problem indexing
- Retrieval of similar cases
- Evaluation of candidate cases
- Modification of selected case, if necessary
- Application to current problem: human action.

The current problem is described by some keywords, attributes or any abstraction that allows describing the basic properties of a case. Based on this a set of close cases description are indexed. The index can be a structure such as for example a classifier or any hierarchical organization of the case base. Among the set of close the closest case cases is determined and is presented as the result of the system. If necessary this case is modified so that it fits to the current problem. The problem solution associated to the current case is applied to the current problem and the result is observed by the user. If the user is not satisfied with the result or no similar case could be found in case base, then case base management starts.

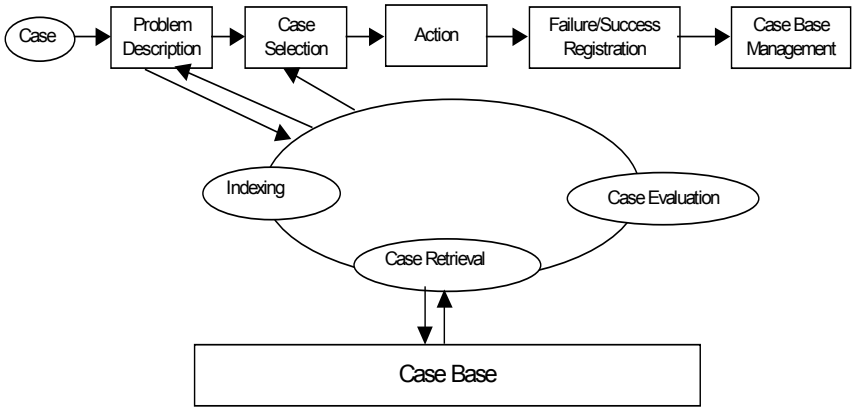


Fig. 1. Case-Based Reasoning Process

2.2 CBR Maintenance

CBR management (see Figure 2) will operate on new cases as well as on cases already stored in case base.

If a new case has to be stored into the case base then it means there is no similar case in case base. The system has recognized a gap in the case base. A new case has to be incorporated into the case base in order to close this gap. From the new case has

to be extracted a predetermined case description, which should be formatted into the predefined case format. Afterwards the case can be stored into case base.

Selective case registration means that no redundant cases will be stored into case base and that the case will be stored at the right place depending on the chosen organization of the case base. Similar cases will be grouped together or generalized by a case that applies to a wider range of problems. Generalization and selective case registration ensure that the case base will not grow too large and that the system can find similar cases fast.

It might also happen that too many cases would be retrieved from case base that are not applicable to the current problem. Then, it might be wise to rethink the case description or to adapt the similarity measure. For the case description, more distinguishing attributes should be found that allow sorting out cases that do not apply to the current problem. The weights in the similarity measure might be updated in order to retrieve only a small set of similar cases.

CBR maintenance is a complex process and works over all knowledge containers (vocabulary, similarity, retrieval, case base) [14] of a CBR system. Consequently, architectures and systems have been developed which support this process [7][15][16].

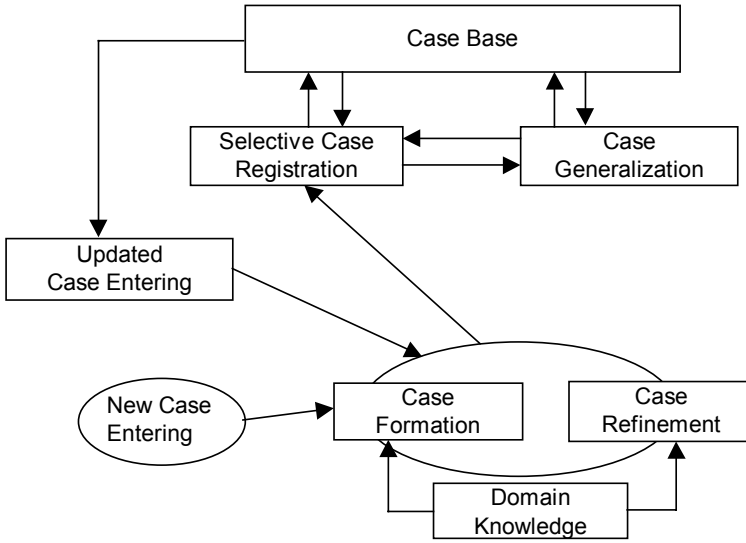


Fig. 2. CBR Maintenance

2.3 Design Consideration

The main problems concerned with the development of a CBR system are:

- What is the right case description?
- What is an appropriate similarity measure for the problem?

- How to organize a large number of cases for efficient retrieval?
- How to acquire and refine a new case for entry in the case base?
- How to generalize specific cases to a case that is applicable to a wide range of situations?

2.4 Case Description

There are different opinions about the formal description of a case. Each system utilizes a different representation of a case. Formally, we like to understand for a case the following definition:

Definition 1 A case F is a triple (P,E,L) with a problem description P, an explanation of the solution E and a problem solutions L.

For image related tasks, we have two main different types of information that make up a case that are image-related information and non-image related information. Image related information could be the 1D, 2D or 3D images of the desired application. Non-image related information could be information about the image acquisition such as the type and parameters of the sensor, and information about the objects or the illumination of the scene. It depends on the type of application what type of information should be taken into consideration for the interpretation of the image. In case of the medical CT image segmentation described in [3] we used patient-specific parameter such as age and sex, slice thickness and number of slices. Jarmulak [1] took into consideration the type of sensor for the railway inspection application. Based on this information the system controls the type of case base that the system is using during reasoning.

How the 2D or 3D image matrix is represented depends on the purpose and not seldom on the developer's point of view. In principle it is possible to represent an image by one of various abstraction levels. An image may be described by the pixel matrix itself or by parts of this matrix (pixel-based representation). It may be described by the objects contained in the image and their features (feature-based representation). Furthermore, it may be described by a more complex model of the image scene comprising of objects and their features as well as the spatial relation between the objects (attributed graph representation or semantic networks).

Jarmular [1] has solved this problem by a four level hierarchy for a case and different case bases for different sensor types. At the lowest level of the hierarchy are stored the objects described by features such as their location, orientation, and type (line, parabola, or noise) parameters. The next level consists of objects of the same channel within the same subcluster. In the following level the subcluster is stored and at the highest level the whole image scene is stored. This representation allows him to match the cases on different granularity levels. Since the whole scene may have distortions caused by noise and imprecise measurements, he can reduce the influence of noise by retrieving cases on these different level.

Grimnes and Aamodt [2] developed a model based image interpretation system for the interpretation of abdominal CT images. The image content is represented by a semantic network where concepts can be general, special cases or, heuristic rules. Not

well understood parts of the model are expressed by cases and can be revised during the usage of the system by the learning component. The combination of the partial well-understood model with cases helps them to overcome the usually burden of modeling. The learning component is based on failure driven learning and case integration. Non-image information is also stored such as sex, age, earlier diagnosis, social condition etc.

Micarelli et. al [4] have also calculated image properties from their images and stored them into the case base. They use the Wavelet transform since it is scale-independent. By doing so they only take into consideration the rotation of the objects in their similarity measure.

In all this work, CBR is only used for the high-level unit. We have studied different approaches for the different processing stages of an image interpretation system. For the image segmentation unit [3], we studied two approaches: 1. a pixel-based approach and 2. a feature-based approach that described the statistical properties of an image. Our results show that the pixel-based approach can give better results for the purpose of image segmentation. For the high-level approach of an ultra sonic image interpretation system, we used a graph-based representation [7].

However, if we do not store the image matrix itself as a case, but we store the representation of a higher-level abstraction instead of, we will lose some information. An abstraction means we have to make a decision between necessary and unnecessary details of an image. It might happen that having not seen all objects at the same time we might think that one detail is not of interest since our decision is only based on a limited number of objects. This can cause problems later on. Therefore, to keep the images themselves is always preferable but needs a lot of storage capacity. The different possible types of representation require different types of similarity measures.

2.5 Similarity

An important point in case-based reasoning is the determination of similarity between a case A and a case B. We need an evaluation function that gives us a measure for similarity between two cases. This evaluation function reduces each case from its case description to a numerical similarity measure *sim*. These similarity measures show the relation to other cases in the case base.

2.5.1 Formalization of Similarity

The problem with similarity is that it has no meaning unless one specifies the kind of similarity.

Smith [17] distinguishes into 5 different kinds of similarity:

- Overall similarity
- Similarity
- Identity
- Partial similarity and
- Partial identity.

Overall similarity is a global relation that includes all other similarity relations. All colloquial similarity statements are subsumed here.

Similarity and identity are relations that consider all properties of objects at once, no single part is left unconsidered. A red ball and a blue ball are similar, a red ball and a red car are dissimilar. The holistic relation's similarity and identity are different in the degree of the similarity. Identity describes objects that are not significantly different. All red balls are similar. Similarity contains identity and is more general.

Partial similarity and partial identity compare the significant parts of objects. One aspect or attribute can be marked. Partial similarity and partial identity are different with respect to the degree of similarity. A red ball and a pink cube are partially similar but a red ball and a red cube are partially identical.

The described similarity relations are in connection with many respects. Identity and similarity are unspecified relations between whole objects. Partial identity and similarity are relations between single properties of objects. Identity and similarity are equivalence relations that mean they are reflexive, symmetrical, and transitive. For partial identity and similarity these relations does not hold. From identity follows similarity and partial identity. From that follows partial similarity and general similarity.

It seems advisable to require from a similarity measure the reflexivity that means an object is similar to itself. Symmetry should be another property of similarity. However, Bayer et. al [18] show that these properties are not bound to belong to similarity in colloquial use. Let us consider the statements "A is similar to B" or "A is the same as B". We notice that these statements are directed and that the roles of A and B can not be exchanged. People say: "A circle is like an ellipse." but not "An ellipse is like a circle." or "The sun looks like the father." but not "The father looks like to the sun.". Therefore, symmetry is not necessarily a basic property of similarity. However, in the above examples it can be useful to define the similarity relation to be symmetrical. The transitivity relation must also not necessarily hold. Let us consider the block world: a red ball and a red cube might be similar; a red cube and a blue square are similar; but a red ball and a blue square are dissimilar. However, a concrete similarity relation might be transitive.

Similarity and identity are two concepts that strongly depend on the context.
The context defines the essential attributes of the objects that are taken into consideration when similarity is determined. An object "red ball" may be similar to an object "red chair" because of the color red. However the object "ball" and "chair" are dissimilar. These attributes may be relevant depending on weather they are given priority or saliency in the considered problem.
This little example shows that the calculation of similarity between the attributes must be meaningful. It makes no sense to compare two attributes that do not make a contribution to the considered similarity.

Since attributes can be numerical and categorical or a combination of both we need to pay attention to this by the selection of the similarity measure. Not all similarity measures can be used for categorical attributes or can deal at the same time with numerical and categorical attributes.

2.5.2 Similarity Measures for Images

Images can be rotated, translated, different in scale, or may have different contrast and energy but they might be considered as similar. In contrast to that, two images may be dissimilar since the object in one image is rotated by 180 degrees. The concept of invariance in image interpretation is closely related to that of similarity. A good similarity measure should take this into consideration.

The classical similarity measures do not allow this. Usually, the images or the features have to be pre-processed in order to be adapted to the scale, orientation or shift. This process is a further processing step which is expensive and needs some a-priori information which are not always given. Filters such as matched filters, linear filters, Fourier or Wavelet filters are especially useful for invariance under translation and rotation which has also been shown by [4]. There has been a lot of work done to develop such filters for image interpretation in the past. The best way to achieve scale invariance from an image is by means of invariant moments, which can also be invariant under rotation and other distortions. Some additional invariance can be obtained by normalization (reduces the influence of energy).

Depending on the image representation (see Figure 3) we can divide similarity measures into:

- pixel (Iconic)-matrix based similarity measures,
- feature-based similarity measures, (numerical or symbolical or mixed type) and,
- structural similarity measures [18]-[23][34].

Since a CBR image interpretation system has also to take into account non-image information such as about the environment or the objects etc, we need similarity measures which can combine non-image and image information. A first approach to this, we have shown in [3].

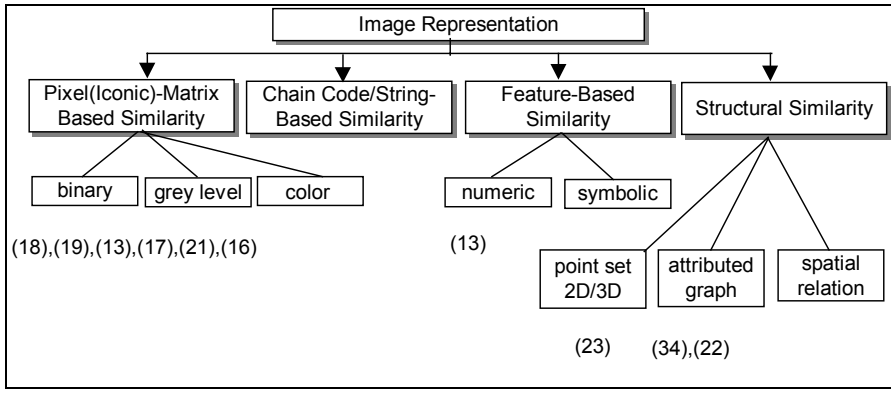


Fig. 3. Image Representations and Similarity Measure

To better understand the concept of similarity systematic studies on the different kinds of image similarity have be done. Zamperoni et. al [19] studied how pixel-matrix based similarity measures behave under different real world influences such as translation, noise (spikes, salt and pepper noise), different contrast and so on. Image

feature-based similarity measures have been studied from a broader perspective by Santini and Jain [20]. Those are the only substantiate works we are aware of. Otherwise at every new conference on pattern recognition new similarity measures [21]-[31] are proposed for specific purposes and the different kinds of image representation but it is missing a more methodological work.

2.6 Organization of Case Base

Cases can be organized into a flat case base or in a hierarchical fashion. In a flat organization, we have to calculate similarity between the problem case and each case in memory. It is clear that this will take time even if the case base is very large. Systems with a flat case base organization usually run on a parallel machine to perform retrieval in a reasonable time and do not allow the case base to grow over a predefined limit. Maintenance is done by partitioning the case base into case clusters and by controlling the number and size of these clusters [33].

To speed up the retrieval process a more sophisticated organization of case base is necessary. This organization should allow separating the set of similar cases from those cases not similar to the recent problem at the earliest stage of the retrieval process. Therefore, we need to find a relation *p* that allows us to order our case base:

Definition: A binary relation p on a set CB is called a partial order on CB if it is reflexive, antisymmetric, and transitive. In this case, the pair ⟨CB, p⟩ is called a partial ordered set or poset.

The relation can be chosen depending on the application. One common approach is to order the case base based on the similarity value. The set of case can be reduced by the similarity measure to a set of similarity values. The relation <= over these similarity values gives us a partial order over these cases. The derived hierarchy consists of nodes and edges. Each node in this hierarchy contains a set of cases that do not exceed a specified similarity value. The edges show the similarity relation between the nodes. The relation between two successor nodes can be expressed as follows: Let z be a node and x and y are two successor nodes of z then x subsumes z and y subsumes z. By tracing down the hierarchy, the space gets smaller and smaller until finally a node will not have any successor. This node will contain a set of close cases. Among these cases is to find the closest case to the query case. Although, we still have to carry out matching the number of matches will have decreased through the hierarchical ordering. The nodes can be represented by the prototypes of the set of cases assigned to the node. When classifying a query through the hierarchy the query is only matched with the prototype. Depending on the outcome of the matching process, the query branches right or left of the node.

Such kind of hierarchy can be created by hierarchical or conceptual clustering [34], k-d trees [35] and decision trees [1]. There are also set-membership based organizations known, such as semantic nets [2] and object-oriented representations [36].

2.7 Learning in a CBR System

CBR management is closely related to learning. It aims to improve the performance of the system.

Let X be a set of cases collected in a case base CB. The relation between each case in case base can be expressed by the similarity value sim. The case base can be partitioned into n case classes C: $CB = \bigcup_{i=1}^{n} C_i$ such that the intra case class similarity is high and the inter case class similarity is low. The set of cases in each class C can be represented by a representative who generally describes the cluster. This representative can be the prototype, the mediod, or an a-priori selected case. Whereas the prototype implies that the representative is the mean of the cluster which can easily be calculated from numerical data. The mediod is the case whose sum of all distances to all other cases in a cluster is minimal. The relation between the different case classes C can be expressed by higher order constructs expressed e.g. as super classes that gives us a hierarchical structure over the case base.
There are different learning strategies that can take place in a CBR system:

1. Learning takes place if a new case x has to be stored into the case base such that: $CB_{n+1} = CB_n \cup \{x\}$. That means that the case base is incrementally updated according to the new case.
2. It may incrementally learn the case classes and/or the prototypes representing the class.
3. The relationship between the different cases or case classes may be updated according the new case classes.
4. The system may learn the similarity measure.

2.7.1 Learning New Cases and Forgetting Old Cases

Learning new cases means just adding cases into the case base upon some notification. Closely related to case adding is case deletion or forgetting cases which have shown low utility. This should control the size of the case base. There are approaches that keep the size of the case base constant and delete cases that have not shown good utility within a fixed time window [37]. The failure rate is used as utility criterion. Given a period of observation of N cases, if the CBR component exhibits M failures in such a period, we define the failure rate as $f_r = M / N$. Other approaches try to estimate the "coverage" of each case in memory and by using this estimate to guide the case memory revision process [38].
The adaptability to the dynamic of the changing environment that requires storing new cases in spite of the case base limit is addressed in [33]. Based on intra class similarity is decided whether a case is to be removed from or to be stored in a cluster.

2.7.2 Learning of Prototypes

Learning of prototypes has been described in [39] for flat organization of case base and for hierarchical representation of case base in [34]. The prototype or the

representative of a case class is the most general representation of a case class. A class of cases is a set of cases sharing similar properties. The set of cases does not exceed a boundary for the intra class dissimilarity. Cases that are on the boundary of this hyperball have maximal dissimilarity value. A prototype can be selected a-priori by the domain user. This approach is preferable if the domain expert knows for sure the properties of the prototype. The prototype can be calculated by averaging over all cases in a case class or the median of the cases is chosen. If only a few cases are available in a class and subsequently new cases are stored in the class then it is preferable to incrementally update the prototype according to the new cases.

2.7.3 Learning of Higher Order Constructs

The ordering of the different case classes gives an understanding of how these case classes are related to each other. For two case classes which are connected by an edge similarity relation holds. Case classes that are located at a higher position in the hierarchy apply to a wider range of problems than those located near the leaves of the hierarchy. By learning how these case classes are related to each other, higher order constructs are learnt [39].

2.7.4 Learning of Similarity

By introducing feature weights we can put special emphasis on some features for the similarity calculation. It is possible to introduce local and global feature weights. A feature weight for a specific attribute is called local feature weight. A feature weight that averages over all local feature weights for a case is called global feature weight. This can improve the accuracy of the CBR system. By updating these feature weights we can learn similarity [40][41].

3 Dissimilarity-Based Classification

Dissimilarity-based pattern recognition (DSC) [8] - also named featureless classification in earlier papers by the authors [42] - means building classifiers based on distance values. Usually, dissimilarity measures can be transformed into similarity measures. Therefore, it could be also named as similarity-based pattern classification. The authors argue that it becomes especially useful when the original data is described by many features or when experts cannot formulate the attributes explicitly, but they are able to provide a dissimilarity measure, instead. Dissimilarity values express a magnitude of difference between two objects and become zero only when the objects are identical. They further argue: Given such a description one does not deal with overlapping classes, provided that distances are truthful representations of the objects. However, exactly the last statement is a crucial point in similarity-based approaches.

DSC works as following: The distance measures between all cases x are calculated. Likewise in hierarchical clustering, the final representation is an $n \times n$ distance matrix. In the learning process, the decision rules are constructed on the complete $n \times n$ pairwise distance matrix, see Figure 4.

A new case is then classified by using their distances to the *n* training cases, see Figure 5. That means a new sample must be compared to all training samples and the dissimilarity measures must be calculated before they are passed to the classifier.

The classifier can be any of the known classification algorithms such as for example a Support-Vector classifier, decision trees, a linear /quadratic classifier, nearest neighbor or Fishers linear discriminant. It has been studied how each classifier performs when the dissimilarity between the objects is calculated based on different similarity measures such as Euclidean distance, Hamming distance, Max-Norm, Box-Cox Transformation, and City Block [9].

Besides the complete *n x n* distance matrices, also their *n x m (m<n)* reduced versions are studied, which are sets of dissimilarities computed between *n* training samples and *m* prototypes chosen from their collection.

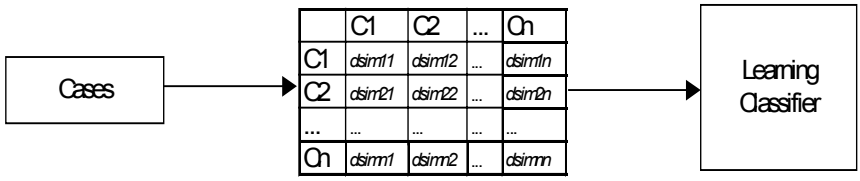


Fig. 4. Learning Dissimilarity-Based Classifier

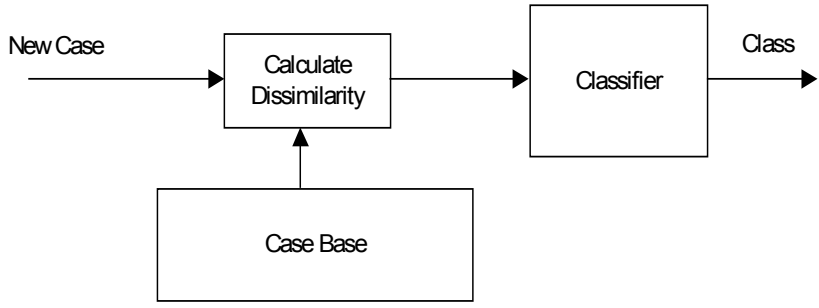


Fig. 5. Dissimilarity-based Classification

The main problems concerned with the development of dissimilarity-based classification are:

- How to access the dissimilarity between the objects?
- What is a proper dissimilarity measure for the problem?
- What is the best type of classifier for the dissimilarity based representation of the objects?
- How to select prototypes?
- What is a representative number of design samples?
- How to organize the system for fast computation?

4. Comparison between CBR and DSC

We have reviewed Case-based Reasoning and Dissimilarity-based Classification. While CBR has been around for more than 10 years, DSC was introduced some years ago. The main focus of the work in DSC is to show that it is possible to built classifiers based on (dis)similarity measures. The study shows that these classifiers do not necessarily work better in terms of accuracy than feature-based classifier [8]. The intention of this work is to overcome the problem of specifying the right image features for classification. Likewise as CBR, DSC relies on the properly chosen similarity measure. The problems with determining similarity have been neglected in the DSC work.

It is argued by Duin et. al [42] that experts are rather able to rank objects based on their dissimilarity instead of describing them by features. However, similarity can have different perspectives as we have shown in Section 2.5. There is no unique way to assess similarity. One person finds two images similar because of the geometric relation between objects in these two images. Another person finds the same images dissimilar since this person does not judge similarity based on the geometric relations between the objects but this person uses the color of the objects in the image to judge similarity. Knowledge engineering experiments for knowledge based image interpretation systems and experiments with repertory grids for determining defect classification knowledge [45] have shown that experts can not easily judge which objects are similar and to what degree they are similar. Also different experts in the field, who are trained to read for example medical images or images showing manufacturing defects, judge similarity of images differently. A consensus of opinion can only be achieved by trying to make explicit the image features and the strategy used by the experts to determine similarity. Therefore, DSC approach does not avoid the knowledge engineering problem; it puts it only in another direction. The assessment of similarity is not a well-understood concept yet. CBR tries to make a step into this direction.

CBR tries to avoid calculation of similarity between all cases and the recent case in order to reduce the computational burden. Therefore, the organization of the case base plays an important role in CBR. The case base should be organized in such a way that similar cases are grouped together and dissimilar cases are separated from them. This should ensure during retrieval of similar cases that such groups of cases that are dissimilar to the recent case are sorted out at an early stage of the retrieval process. This organization is based on the similarity relation between the cases in the case base. The recent case is classified through the organization structure based on its similarity to the cases in the case base. The organization of the case base is related to the classification in DSC. The classifiers in DSC also try to find the boundaries between the subspace of similar cases. While the calculation of similarity between the recent case and the cases in the case base stays explicit during the classification in CBR, in DSC this calculation must be carried out before the recent case is given to the classifier. The computational burden in DSC is enormous even for small case bases.

CBR has been introduced by the artificial intelligence community. Naturally, this community focuses on methods which make knowledge explicit. The assessment of similarity should stay explicit to the user in order to understand the concept of

similarity better. Under this requirement, classifiers such as support vector machines, linear discriminate analysis are not sufficient. Following the trend in pattern recognition which relies on numbers instead of on symbolic knowledge, the classifiers are different in DSC from those in CBR.

DSC has similarities to hierarchical clustering [44]. In hierarchical clustering the $n \times n$ similarity matrix is also used and based on this similarity matrix hierarchical groups of similar cases are calculated. While in clustering the classification rules is not made explicit, in DSC the rules are learnt by the used classifier. Conceptual clustering [43] are methods which make the classification rules explicit. To this respect DSC is similar to conceptual clustering. However, conceptual clustering explains the way similarity has been accessed and does not require the calculation of similarity beforehand. In DSC the similarity of the actual object to all cases in the case base must always be calculated before the classification process.

Conceptual clustering methods are used to built index trees for CBR systems [34][35]. They are always used in an incremental fashion in order to update them according to new acquired cases. DSC does not consider the aspect of incremental learning. Learning is only understood as learning of classifier from the initial similarity matrix. DSC does not consider the different types of learning such as learning of new cases; prototype learning and learning of similarity which are necessary to ensure that the system will improve their performance. It is assumed that such kind of classifiers can be built on sets with small sample size [9]. This might be true if the sample set is a good representative of the domain. However, it has been shown in CBR that maintenance of the case base is an important issue.

CBR community has focussed on all aspects of CBR from basic principles to software engineering aspects and developed a lot of good ideas that have been shown excellent performance in practice. The work on DSC is preliminary and does not consider the engineering aspect. Many topics that have been worked out in CBR are relevant for DSC such as how to define similarity, incremental learning, prototype selection, software engineering aspects and so on.

Finally, we think that DSC is only a variant of CBR and that DSC can benefit from the concepts developed in CBR.

5. Conclusion

We have compared Case-based Reasoning and Dissimilarity-based Classification. Both approaches use the (dis)similarity measure between the new case and cases in the cases base to classify the new case. The difference between CBR and DSC is that in DSC the (dis)similarity measure between the new case and all cases in the case base must be calculated before the classification. It is clear that such an approach is computationally expensive. The classification algorithms used in DSC are traditional pattern recognition algorithms such as support vector machines, linear discriminat function and decision trees. The assessment of similarity stays always explicit during the reasoning process in CBR. Traditionally this community tries to develop methods that have explanation capability.

While CBR considers all aspects of the similarity based reasoning the work on DSC does not. Finally, we think that DSC can learn a lot from CBR.

Acknowledgement

We would like to thank Maria Petrou for her kind advice on this topic.

References

1. Jarmulak, Case-Based Classification of Ultrasonic B-Scans: Case-Base Organisation and Case Retrieval, In: B. Smyth and P. Cunningham, Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 100-111.
2. M. Grimnes and A. Aamodt, A Two Layer Case-Based Reasoning Architecture for Medical Image Understanding, In: I. Smith and B. Faltings (Eds.), Advances in Case-Based Reasoning, lnai 1168, Springer Verlag 1996, pp 164-178.
3. P. Perner, An Architeture for a CBR Image Segmentation System, Journal on Engineering Application in Artificial Intelligence, Engineering Applications of Artificial Intelligence, vol. 12 (6), 1999, p. 749-759.
4. A. Micarelli, A. Neri, and G. Sansonetti, A Case-Based Approach to Image Recognition, In: E. Blanzieri and L. Portinale (Eds.), Advances in Case-Based Reasoning, lnai 1898, Springer Verlag 2000, p. 443-454.
5. W. Cheetham and J. Graf, Case-Based Reasoning in Color Matching, In: Leake, D.B. and Plaza, E. (Eds.) Case-Based Reasoning Research and Development, Springer Verlag 1997, 1-12.
6. V. Ficet-Cauchard, C. Porquet, and M. Revenu, CBR for the Reuse of Image Processing Knowledge: A Recursive Retrieval/Adaption Strategy, In: K.-D. Althoff, R. Bergmann, and L. Karl Branting (Eds.), Case-Based Reasoning Research and Development, 1999, lnai 1650, p. 438-453.
7. P. Perner, Using CBR Learning for the Low-Level and High-Level Unit of a Image Interpretation System, ICAPR´98, Plymouth, peer reviewed conference, In: Sameer Singh (Eds.), Advances in Pattern Recognition, Springer Verlag 1998, p. 45-54.
8. E. Pekalska and R.P.W. Duin, Classifier for dissimilarity-based pattern recognition, In Proc.: A. Sanfeliu et. al (Eds.), 15th Intern. Conference on Pattern Recognition, Barcelona 2000, IEEE Computer Society PR 00750, p. 12-16.
9. R. Duin, Classifiers in Almost Empty Spaces, In Proc.: A. Sanfeliu et. al (Eds.), 15th Intern. Conference on Pattern Recognition, Barcelona 2000, IEEE Computer Society PR 00750, p. 1-7.
10. E. Blanzieri and L. Portinale (Eds.), Advance in Case-Based Reasoning, Springer Verlag, lnai 1898, 2000.
11. K.-D. Althoff, R. Bergmann, and L.K. Branting (Eds.), Case Based Reasoning Research and Development, Springer Verlag, lnai 1650, 1999.
12. K.-D. Althoff, Case-Based Reasoning, In: S.K. Chang (ed.) Handbook of Software Engineering and Knowledge Engineering, vol. I, World Scientific (to appear).
13. Wess St., Globig Chr. Case-Based and Symbolic Classification. In: Wess St., Althoff K.-D., Richter M.M. (eds.). Topics in Case-Based Reasoning. Springer Verlag 1994, pp 77-91.
14. M.M. Richter, Introduction (to Case-Based Reasoning), In: M. Lenz et. al (Eds.) Case-Based Reasoning Technology: From Foundations to Applications, Springer Verlag 1998, lnai 1400.

15. F. Heister and W. Wilke, An Architecture for Maintaining Case-Based Reasoning Systems, In: B. Smyth and P. Cunningham (Eds.), Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 221-232.
16. J. Lluis Arcos and E. Plaza, A reflective Architecture for Integrated Memory-Based Learning and Reasoning, In: St. Wess, K.-D. Althoff, and M.M. Richter (Eds.) Topics in Case-based Reasoning, Springer Verlag 1993, p. 289-300.
17. L.B. Smith, From global similarities to kinds of similarities: the construction of dimensions in development. In: St. Vosniadou and A. Ortony (Eds.), Similarity and Analogical Reasoning, Cambridge University Press, 1989
18. M. Bayer, B. Herbig, and St. Wess, Similarity and Similarity Measures, In: S. Wess, K.D. Althoff, F. Maurer, J. Paulokat, R. Praeger, and O. Wendel (Eds.), Case-Based Reasoning Bd. I, SEKI WORKING PAPER SWP-92-08 (SFB)
19. P. Zamperoni and V. Starovoitov, „How dissimilar are two gray-scale images“, In Proc. of 17. DAGM Symposium 1995,Springer Verlag, pp.448-455
20. S. Santini and R. Jain, Similarity Measures, IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 21, No. 9,1999, pp. 871-883
21. Y. Horikowa, Pattern Recognition with Invariance to Similarity Transformations Based on Third-Order Correlations, In Proceedings of IAPR´96, Volume II, Track B, pp 200-204
22. F. Leitao, A Study of String Dissimilarity Measures in Structural Clustering, In: S. Singh (Eds.), Advances in Pattern Recognition, Springer Verlag 1999, pp 385-394.
23. G. Mehrotra, Similar Shape Retrieval Using a Structural Feature Index, Information Systems, vol. 18 (5), 1993, pp. 525-537.
24. C. Cortelazzo, G. Deretta, G.A., Mian, P. Zamperoni, Normalized weighted Levensthein distance and triangle inequality in the context of similarity discrimination of bilevel images, Pattern Recognition Letters, vol. 17, no. 5, 1996, pp. 431-437
25. A. Crouzil, L. Massipo-Pail, S. Castan, A New Correlation Criterion Based on Gradient Fields Similarity, In Proceedings of IAPR´96, vol. I, Track A, p. 632-636
26. Moghadda, Nastar, Pentland, A Bayesian Similarity Measure for Direct Image Matching, In Proc. of ICPR´96, vol. II, Track B, pp. 350-358.
27. Moghadda, Jebra, Pentland, Efficient MAP/ML Similarity Matching for Visual Recognition, In Proc. of ICPR´98, vol. I, pp. 876-881
28. Wilson, Baddely, Owens, A new metric for gray-scale image comparison, Intern. Journal of Computer Vision, vol. 24, no. 1, pp. 5-19
29. B. Messmer and H. Bunke, Efficient subgraph isomorphism detection: a decomposition approach, IEEE Trans. on Knowledge and Data Engineering, vol 12, No. 2, 2000, pp. 307-323
30. A. van der Heiden and A. Vossepoel A Landmark-Based Approach of Shape Dissimilarity, In Proc. of ICPR 1999, vol. I, Track A, pp. 120-124
31. P. Perner, Content-Based Image Indexing and Retrieval in a Image Database for Technical Domains, In: Multimedia Information Analysis and Retrieval, Horace H.S. Ip and A. Smuelder (Eds.), LNCS 1464, Springer Verlag 1998, p. 207-224
32. A. Voß (Eds.), Similarity Concepts and Retrieval Methods, Fabel Report No. 13, 1993, ISSN 0942-413X
33. J. Surma and J. Tyburcy, A Study on Competence-Preserving Case Replacing Strategies in Case-Based Reasoning, In: B. Smyth and P. Cunningham (Eds.), Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 233-238.
34. P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.
35. St. Wess, K.-D. Althoff, and G. Derwand, Using k-d Trees to Improve the Retrieval Step in Case-Based Reasoning, In: St. Wess, K.-D. Althoff, and M.M. Richter (Eds.) Topics in Case-based Reasoning, Springer Verlag 1993, p. 167-182.

36. R. Bergmann and A. Stahl, Similarity Measures for Object-Oriented Case Representations, In Proc.: Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, p. 25-36.
37. L. Portinale, P. Torasso, and P. Tavano, Speed-Up, Quality and Competence in Multi-modal Case-Based Reasoning, In: K.-D. Althoff, R. Bergmann, and L. K. Branting (Eds.) Case-Based Reasoning Research and Development, lnai 1650, Springer Verlag 1999, p. 303-317.
38. B. Smyth and E. McKenna, Modelling the Competence of Case-Bases, In: B. Smyth and P. Cunningham (Eds.), Advances in Case-Based Reasoning, lnai 1488, Springer Verlag 1998, p. 208-220.
39. Perner P., Paetzold W. An Incremental Learning System for Interpretation of Images. In: D. Dori and A. Bruckstein (eds.). Shape, Structure, and Pattern Recognition. World Scientific Publishing Co., 1995, pp 311-323.
40. D. Wettscherek, D.W. Aha and T. Mohri, A review and empirical evaluation of feature weighting methods for a class of lazy learning algorithms, in Artifical Intelligence Review (also available on the Web from http://www.aic.nrl.navy.mil/~aha.
41. A. Bonzano and P. Cunningham, Learning Feature Weights for CBR: Global versus Local
42. R.P.W. Duin, D. de Ridder, and D.M.J. Tax, Featureless Classification, Kybernetica, vol. 34, no. 4, 1998, p. 399-404.
43. G. Briscoe and T. Caelli, A Compendium of Machine Learning, Vol. 1: Symbolic Machine Learning, Ablex Publishing Corporation , Norwood, New Jersey, 1996
44. A.K. Jain and R.C. Dubes Algorithm for Clustering Data, Prentice Hall 1998
45. P. Perner, How to use Repertory Grid for Knowledge Acquisition in Image Interpretation. HTWK Report 2, 1994.

FAM-Based Fuzzy Inference for Detecting Shot Transitions

Seok-Woo Jang, Gye-Young Kim, and Hyung-Il Choi

Soongsil University, 1-1, Sangdo-5 Dong, Dong-Jak Ku, Seoul, Korea
swjang@vision.soongsil.ac.kr, gykim,hic@computing.soongsil.ac.kr

Abstract. We describe a fuzzy inference approach for detecting and classifying shot transitions in video sequences. Our approach basically extends FAM(Fuzzy Associative Memory) to detect and classify shot transitions, including cuts, fades and dissolves. We consider a set of feature values that characterize differences between two consecutive frames as input fuzzy sets, and the types of shot transitions as output fuzzy sets. An initial implementation runs at approximately 7 frames per second on PC and yields promising results.

1 Introduction

The amount of digital video that is available has increased dramatically in the last few years. For automatic video content analysis and efficient browsing, it is necessary to split the sequences of video into more manageable segments. The transition of camera shot may be a good candidate for such a segmentation [1]. Shot transitions may be classified into three major types [2]. A cut is an instantaneous transition from one scene to the next. A fade is a gradual transition between a scene and a constant image (fade out) or between a constant image and a scene (fade in). A dissolve is a gradual transition from one scene to another, in which the first scene fades out and the second scene fades in. Typically, fade out and fade in begin at the same time, and the fade rate is constant.

Many researchers, especially in multimedia community, are working on shot transition detection, and different detection techniques have been proposed and continue to appear [3][4][5]. But, most of the existing techniques seem to work in restricted situations and lack robustness, since they use simple decision rules like thresholding and they heavily rely on intensity based features like histograms. We believe that the detection of a shot transition intrinsically involves the nature of fuzzyness, as many transitions occur quite gradually and decision about the transitions should consider rather compound aspects of image appearance.

This paper presents a fuzzy inference approach for detecting shot transitions, which basically extends FAM(Fuzzy Associative Memory). FAM provides a framework which maps one family of fuzzy sets to another family of fuzzy sets [6]. This mapping can be viewed as a set of fuzzy rules which associate input fuzzy sets with output fuzzy sets. We consider a set of feature values that characterize differences between two consecutive frames as input fuzzy sets, and the

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 52–61, 2001.

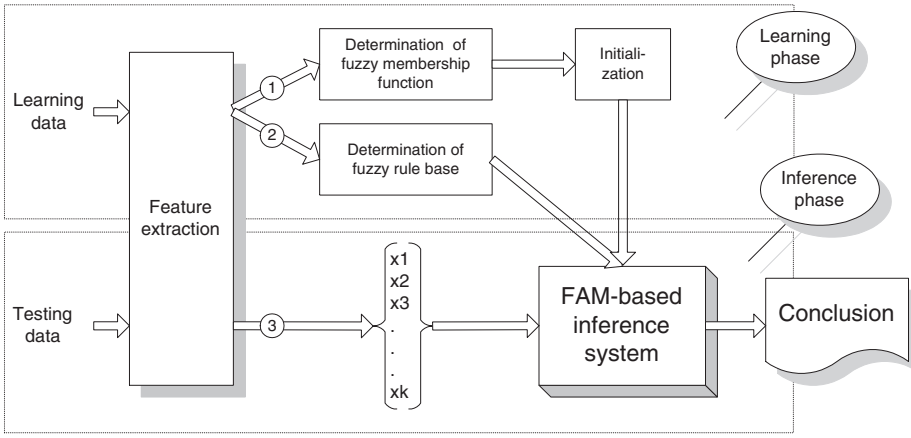


Fig. 1. System Organization

types of shot transitions as output sets. Figure 1 shows the overall organization of the inference system.

Our inference system consists of three main parts; a feature extraction part, learning part and inferring part. The feature extraction part is common to both of learning and inferring part. It compares two consecutive frames and computes predefined feature values. The features are to evaluate chromatic changes between two consecutive frames, which reflect clues about shot transitions. The details are discussed in section 2. The learning part analyzes learning video data made up of input and output pairs in order to form fuzzy sets. It then generates a correlation matrix which shows the degree of association between input and output fuzzy sets. The details of the learning part will be discussed in section 4. The inferring part processes test video data and draws conclusions with the model built up in the learning part. The details of the inferring part will be discussed in section 3.

2 Feature Set

We use HSI color model to represent a color in terms of hue, saturation and intensity. So the first step of feature extraction is to convert RGB components of a frame into HSI color representation. Our feature set contains three different types of measures on frame differencing and changes in color attributes. The first one is the correlation measure of intensities and hues between two consecutive frames. If two consecutive frames are similar in terms of the distribution of intensities and hues, it is very likely that they belong to a same scene. This feature is very easy to compute. But it works reasonably well, especially for detecting a cut. Figure 2 shows the procedure of computing the feature.

We compute area-wise correlation rather than simple differences between consecutive frames. The area-wise operation is to reduce the influence of noises,

and the correlation operation is to reflect the distribution of values.

$$F_{Corr} = \alpha \times Corr(BIM_{t-1}, BIM_t) + \beta \times Corr(BHM_{t-1}, BHM_t) \quad (1)$$
$$where \quad 0 \leq F_{Corr} \leq 1, \quad 0 \prec \alpha \leq \beta \prec 1, \quad \alpha + \beta = 1$$

In (1), BIM(Block Intensity Mean) denotes the average of block intensities and BHM(Block Hue Mean) denotes the average of block hue values. The α and β are weighting factors that control the importance of related terms. We assign a higher weight to β, as hues are less sensitive to illumination than intensities are.

The second feature is to evaluate how intensities of successive frames vary in the course of time. This feature is especially useful for detecting fades. During a fade, frames have their intensities multiplied by some value of α. A fade in forces α to increase from 0 to 1, while a fade out forces α to decrease from 1 to 0. In other words, overall intensities of frames transit toward a constant. To detect such a variation, we define a ratio of overall intensity variations as in (2).

$$F_{Diff} = \frac{D_{Idiff}}{D_{IAdiff}} \quad (2)$$

$$D_{Idiff} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\big(I(i,j,t) - I(i,j,t-1)\big)}{K \cdot M \cdot N}$$

$$D_{IAdiff} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\big|\, I(i,j,t) - I(i,j,t-1)\big|}{K \cdot M \cdot N}$$

$$where\ k = intensity\ level,$$
$$M,\ N = frame\ height,\ width$$

The above ratio ranges from -1 to 1, revealing whether frames become bright or dark. It has negative values during a fade out and positive values during a fade in, while the magnitude of the values approaches to 1. On the other hand, the ratio remains unvaried during a normal situation.

The third feature evaluates the difference of differences of saturations along a sequence of frames. This feature is especially useful for detecting a dissolve, since its behavior resembles that of a laplacian(a second derivative). A laplacian usually reveals a change of the direction of variations. A dissolve occurs when one scene fades out and another scene fades in. In other words, the direction of fades switches at the time of a dissolve. Furthermore, in order to make a smooth transition at a dissolve instant, the overall saturation of frames tends to become low. Therefore, the values of (3) crosses a zero at an instant of a dissolve.

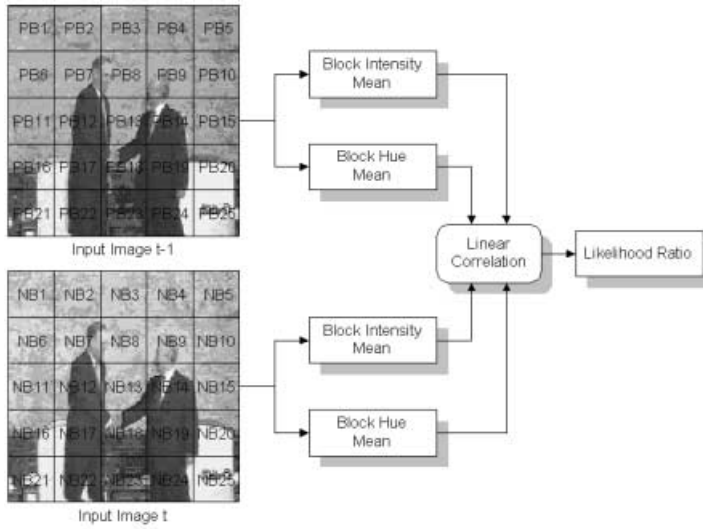


Fig. 2. A feature of block correlation

$$F_{Laplacian} = S_t - S_{t-1} \tag{3}$$

$$S_t = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\big(S(i,j,t) - S(i,j,t-1)\big)}{K \cdot M \cdot N}$$

$$S_{t-1} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N}\big(S(i,j,t-1) - S(i,j,t-2)\big)}{K \cdot M \cdot N}$$

The table 1 summarizes the behavioral characteristics of our features. The "stay" column of the table denotes the case where shot transitions do not occur. As noted, each features show some distinct values for various types of shot transitions. Such discriminating abilities are to be sorted and organized in terms of inference mechanism in section 3 and section 4.

Table 1. Behavioral characteristics of features

type	stay	cut	fade in	fade out	dissolve
F_{Corr}	high	low	high	high	high
F_{Diff}			close to 1	close to -1	
$F_{Laplacian}$					0-crossing

3 Inferring Model

The extracted feature values are fed into the inference mechanism for detecting and classifying shot transitions(including cuts, fades and dissolves) in digital video sequences. We suggest a fuzzy inference system which employs FAM for implementing fuzzy rules. That is, we interpret an input associant as an antecedent part of a fuzzy rule, an output associant as a consequent part, and a synaptic weight as the degree of reliability of the rule. Figure 3 shows the structure of our inferring model which consists of five layers. We have 3 input variables $x_i(F_{Corr},\ F_{Diff},\ F_{Laplacian})$ and one output variable y. Each input variable x_i furnishes p_i fuzzy sets, and the output variable furnishes m fuzzy sets.

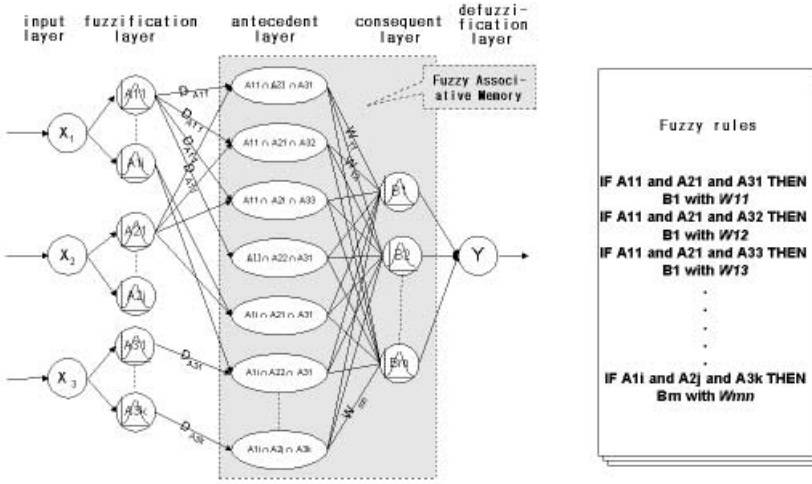


Fig. 3. Model of FAM based fuzzy inference system

The input layer of Figure 3 just accepts input feature values. Thus, the number of nodes in the input layer becomes 3. The fuzzification layer contains membership functions of input features. The output of this layer then becomes the fit values of input to associated membership functions.

The antecedent layer contains antecedent parts of fuzzy rules, which have the form of logical AND of individual fuzzy terms. We allow every possible combinations of fuzzy sets drawn one from each group of p_i fuzzy sets. Each incoming link has a weight which represents the degree of usefulness of an associated fuzzy set. If links from some node of the fuzzification layer have a high value of weight, it means that the fuzzy set contained in the node is very useful in inferring a desired conclusion. Each node of this layer just compares incoming weighted values and takes the minimum of them.

The consequent layer contains consequent parts of fuzzy rules. This layer contains 5 membership functions(stay, cut, fade in, fade out, dissolve) of an output variable. We allow full connections between the antecedent layer and the consequent layer. But, each connection may have a different value of weight, which represents the degree of credibility of each connection. We basically follow the max-min compositional rule of inference [7]. Thus, when N antecedent nodes $A_1, ..., A_N$ are connected to the j-th consequent node B_j with weight w_{ij}'s, the output of the j-th consequent node becomes a fuzzy set whose membership function is defined as in (4).

$$\mu'_{B_j}(y) = min\left[\max_{1\le i\le N}\Big\{min\big(w_{ij}, output(A_i)\big)\Big\}, \mu_{B_j}(y)\right] \tag{4}$$

where $\mu_{B_j}(y)$ is a membership function contained in the j-th consequent node, and $output(A_i)$ is an output of the i-th antecedent node. The output of each consequent node has the form of fuzzy set. The defuzzification layer then combines incoming results which are in the form of fuzzy sets, and produces a final crisp conclusion. Here we use a centroidal defuzzification technique which computes the center of mass of incoming fuzzy sets [7]. That is, the final output y^* is computed as in (5).

$$y^* = \frac{\sum_{y_i} y_i \cdot \Big\{\max_k\big[\mu'_{B_k}(y_i)\big]\Big\}}{\sum_{y_i}\Big\{\max_k\big[\mu'_{B_k}(y_i)\big]\Big\}} \tag{5}$$

4 Learning Model

Our inferring model can work properly only when membership functions as well as synaptic weights are determined in advance. In this section, we propose a learning method which derives the necessary information from given input-output learning data. This section has two main parts. The first part is determination of the number of fuzzy sets for each variable and corresponding membership functions. The second part is the determination of synaptic weights.

The first problem associated with fuzzy inference is how to divide the range of each input and output variable into how many subranges [8]. We then have to associate each subrange with a proper membership function. We solve such problems by analyzing histograms of each input and output variables. We first define a basic structure of a membership function as in Figure 4 which is a mixture of trapezoidal and sigmod functions. A membership function will then be refined later by tuning the structure to a constructed histogram. The basic structure G has five parameters as in (6), and these parameters form three basic functions; left and right sigmoid functions, and central base function with a value of 1.

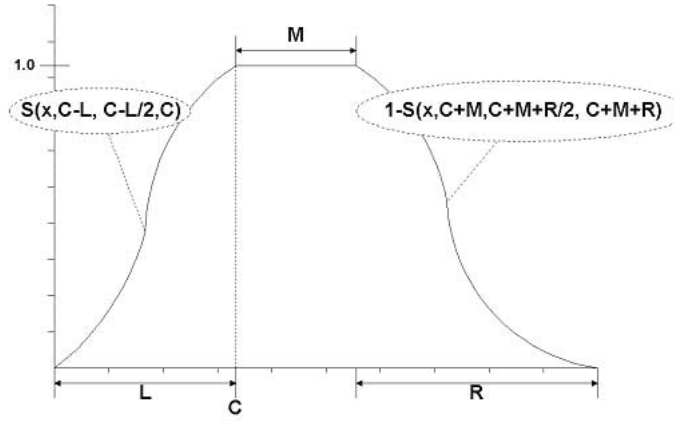


Fig. 4. Basic structure of membership function

$$G(x, C, L, R, M) = \tag{6}$$
$$\begin{cases} S(x, C-L, C-L/2, C) & \text{if} \quad x \prec C \\ 1 & \text{if} \quad C \leq x \prec C+M \\ 1 - S(x, C+M, C+M+R/2, C+M+R) & \text{if} \quad x \geq C+M \end{cases}$$

$$s(x, \alpha, \beta, \gamma) = \begin{cases} 0 & \text{if} \quad x \prec \alpha \\ 2\big((x-\alpha)(\gamma-\alpha)\big)^2 & \text{if} \quad \alpha \leq x \prec \beta \\ 1 - 2((x-\gamma)/(\gamma-\alpha))^2 & \text{if} \quad \beta \leq x \prec \gamma \\ 1 & \text{if} \quad x \geq \gamma \end{cases} \tag{7}$$

In (6), x is a variable on which a membership function is to be defined, M denotes the length of a central base which has a value of 1. L and R denotes the left and right range on which a left and right sigmoid function is to be defined, respectively. One important characteristic of the structure G is that the left and right sigmoid functions as well as the central base function can be adjusted independently. Thus, we can have diverse forms of membership functions by changing the five parameters.

We have formed prototypical membership functions for each input and output variables. We now define a measure which represents the degree of usefulness of input fuzzy sets. When the range of a variable on which a fuzzy set is defined contains learning data whose output values have a homogeneous nature, we may say that the fuzzy set is very useful in deriving a desired conclusion. We use the degree of homogeneity of the output values as an indicator to usefulness of a relevant input fuzzy set.

As another important factor for determining the usefulness of fuzzy sets, we may consider the amount of separateness between adjacent fuzzy sets. When a fuzzy set is well separated from its neighboring fuzzy sets defined on the same input variable, we may say the fuzzy set is meaningful and also useful in deriving a desired conclusion. Based on the above conjectures, we define $D_{i,j}$, the degree of usefulness of the j-th fuzzy set of the i-th input feature, as in (8).

$$D_{i,j} = 1 - \left[\frac{1}{N_i - 1} \sum_{k,k \neq j} \frac{area(G_{i,j} \cap G_{i,k})}{area(G_{i,j})} \right] \times H_{i,j} \tag{8}$$

$$H_{i,j} = 1 - \frac{number\ of\ O(G_{i,j})\ in\ major\ class}{total\ number\ of\ O(G_{i,j})}$$

In (8), N_i is the number of fuzzy sets defined on the i-th input feature and $O(G_{i,j})$ denotes outputs which are associated with inputs belong to $G_{i,j}$. $D_{i,j}$ has two major components. The first component considers the amount of overlaps between $G_{i,j}$ and its neighboring membership functions. As this overlap becomes smaller, $D_{i,j}$ gets closer to a value of 1. But if this overlap becomes larger, $D_{i,j}$ gets closer to 0. The second component evaluates the homogeneity of $O(G_{i,j})$. In fact, we count the number of $O(G_{i,j})$ whose class index is a major one and divide the counted number by the total number of $O(G_{i,j})$.

Our inference system also requires a predetermined correlation matrix which represents the degrees of associations between input and output fuzzy sets. We take a Hebbian-style learning approach to build up the correlation matrix. The Hebbian learning is an unsupervised learning model whose basic idea is that "the synaptic weight is increased if both an input and output are activated [9]." We take input and output values as fit values to membership functions. Thus, when $a_i(n)$ is an input associant for the n-th learning datum and $b_j(n)$ is an output associant for the n-th learning datum, the change of weight is carried out as in (9).

$$w_{ij}(n) = w_{ij}(n-1) \oplus \eta \cdot a_i(n) \otimes b_j(n) \tag{9}$$

In (9), η is a positive learning rate which is less than 1. This learning rate controls the average size of weight changes. $\otimes$ represents a minimum operator and $\oplus$ represents a maximum operator. If we denote the n-th output vector of the antecedent layer as X and the n-th output vector of the consequent layer as Y, our correlation matrix can be learned iteratively as in (10).

$$\begin{aligned} W(n) &= W(n-1) \oplus \eta \cdot \Delta W(n) \\ &= W(n-1) \oplus \eta \cdot (X^T \otimes Y) \end{aligned} \tag{10}$$

In (10), the output vector of each layer corresponds to fit values of a learning datum to membership functions which resides in the layer. The encoded correlation matrix together with associated membership functions represents a set of fuzzy rules.

5 Experimental Results and Conclusions

The proposed approach that detects shot transitions by the output of the fuzzy inference mechanism is applied to mpeg files. The files include music videos, movies, news and advertisements. The total number of frames is 7814. They

Table 2. Accuracy of shot transition detection

Method	cut			fade in(out)			dissolve		
	N_c	N_m	N_f	N_c	N_m	N_f	N_c	N_m	N_f
intensity histogram	55	10	7	5(4)	3(4)	3(2)	4	4	2
edge	57	8	6	7(7)	1(1)	2(3)	5	3	2
motion vector	61	4	3	5(6)	3(2)	4(3)	4	4	3
proposed method	65	0	0	8(8)	0(0)	0(0)	6	2	1

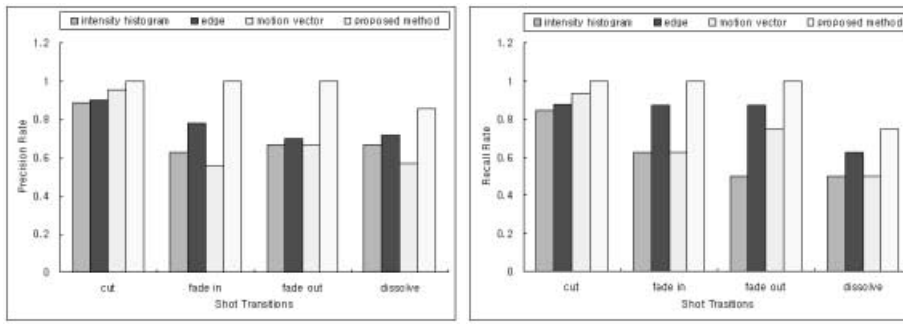
include 65 cuts, 8 fades and 8 dissolves. We compared the performance of our approach against those of such approaches as the intensity histogram difference [3], the edge counting [4], and the motion vector detection [5]. Table 2 summarizes the comparison in terms of accuracy, where N_c denotes the number of shot transitions that are correctly detected, N_m denotes the number of misses, and N_f denotes the number of false positives. Our approach was able to detect all of cuts and fades with no false positives or misses. For the dissolve transitions, we had 2 misses and 1 false positive.

We also evaluated the performance in terms of "precision rate" and "recall rate". The precision rate depicts the ratio of the number of correctly detected transitions against the total number of declared transitions, while the recall rate expresses the ratio of the number of correctly detected transitions against the total number of actual transitions

$$R_{precision} = \frac{N_c}{N_c + N_f} \tag{11}$$

$$R_{recall} = \frac{N_c}{N_c + N_m} \tag{12}$$

Figure 5 summarizes the comparison in terms of the precision rate and recall rate. We can note that our approach outperforms others for every type of transitions in both criteria.



(a) Precision rate (b) Recall rate

Fig. 5. Precision rate and recall rate

In our experiments, the learning rate η in (10) was set to 1.0 and the initial weights $W(0)$ in (10) were set to 0. We examined convergence rates of synaptic weights by changing values of the learning rate and the initial weights and noticed that the values do not affect the performance seriously. To sum up, our fuzzy inference approach seems to work as a promising solution for detecting shot transitions, even though results obviously depend on the involved features. One distinct merit of our inference mechanism is that it can combine various types of features into integrated fuzzy rules and automatically attach the measure of importance to each features. Such a measure can treat involved features discriminately to lead to more accurate conclusions.

References

1. Ahmed K. Elmagarmid, Abdelsalam A. Helal, Anupam Joshi, and Magdy Ahmed: Video Database Systems. Kluwer Academic Publishers (1997)
2. Hong Heather Yu and Wyne Wolf: Multi-resolution video segmentation using wavelet transformation. Storage and Retrieval for Image and Video Databases, SPIE Vol. 3312 (1998) pp. 176-187
3. H. J. Zhang, A. Kankanhalli, and S. W. Smoliar: Automatic partitioning of full-motion video. Proc. of ACM Multimedia Systems, Vol. 1 (1993) pp. 10-28
4. Yi Wu and David Suter: A comparison of methods for scene change detection in noisy image sequence. First International Conference on Visual Information Systems (1996) pp. 459-468
5. Ramin Zabih, Justin Miller, and Kevin Mai: Feature-based algorithms for detecting and classifying scene breaks. Technical Report CS-TR-95-1530, Cornell University Computer Science Department (1995)
6. Kosko B: Neural Network and Fuzzy Systems. Prentice-Hall International (1994)
7. Zimmermann HJ: Fuzzy Set Theory and Its Applications. KALA (1987)
8. Hideyuki T and Isao H: NN-driven fuzzy reasoning. International Journal of Approximate Reasoning (1991) pp. 191-212
9. Freeman JA and Skapura DM: Neural Networks - Algorithms, Applications and Programming Techniques. Addison Wesley Publishing Company (1991)

Rule-Based Ensemble Solutions for Regression

Nitin Indurkhya and Sholom M. Weiss

IBM T.J. Watson Research Center,
P.O. Box 218, Yorktown Heights, NY 10598, USA
nitin@data-miner.com, sholom@us.ibm.com

Abstract. We describe a lightweight learning method that induces an ensemble of decision-rule solutions for regression problems. Instead of direct prediction of a continuous output variable, the method discretizes the variable by k-means clustering and solves the resultant classification problem. Predictions on new examples are made by averaging the mean values of classes with votes that are close in number to the most likely class. We provide experimental evidence that this indirect approach can often yield strong results for many applications, generally outperforming direct approaches such as regression trees and rivaling bagged regression trees.

1 Introduction

Prediction methods fall into two categories of statistical problems: classification and regression. For classification, the predicted output is a discrete number, a class, and performance is typically measured in terms of error rates. For regression, the predicted output is a continuous variable, and performance is typically measured in terms of distance, for example mean squared error or absolute distance.

In the statistics literature, regression papers predominate, whereas in the machine learning literature, classification plays the dominant role. For classification, it is not unusual to apply a regression method, such as neural nets trained by minimizing squared error distance for zero or one outputs. In that restricted sense, classification problems might be considered a subset of regression methods.

A relatively unusual approach to regression is to discretize the continuous output variable and solve the resultant classification problem. In (Weiss & Indurkhya, 1995), a method of rule induction was described that used k-means clustering to discretize the output variable into classes. The classification problem was then solved in a standard way, and each induced rule had as its output value the mean of the values of the cases it covered in the training set. A hybrid method was also described that augmented the rule representation with stored examples of each rule, resulting in reduced error for a series of experiments.

Since that earlier work, very strong classification methods have been developed that use ensembles of solutions and voting (Breiman, 1996; Bauer & Kohavi, 1999; Cohen & Singer, 1999; Weiss & Indurkhya 2000). In light of the

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 62–72, 2001.

newer methods, we reconsider solving a regression problem by discretizing the continuous output variable using k-means and solving the resultant classification problem. The mean or median value for each class is the sole value to be stored as a possible answer when that class is selected as an answer for a new example.

To test this approach, we use a recently developed, lightweight rule induction method (Weiss & Indurkhya, 2000). It was developed strictly for classification, and like other ensemble methods performs exceptionally well on classification applications. However, classification error can diverge from distance measures used for regression. Hence, we adapt the concept of margins in voting for classification (Schapire et al., 1998) to regression where, analogous to nearest neighbor methods for regression, class means for close votes are included in the computation of the final prediction.

Why not use a direct regression method instead of the indirect classification approach? Of course, that is the mainstream approach to boosted and bagged regression (Friedman et al., 1998). Some methods, however, are not readily adaptable to regression in such a direct manner. Many rule induction methods, such as our lightweight method, generate rules sequentially class by class. Why not try a trivial preprocessing step to discretize the predicted continuous variable? Moreover, if good results can be obtained with a small set of discrete values, then the resultant solution can be far more elegant and possibly more interesting to human observers. Lastly, just as experiments have shown that discretizing the input variables may be beneficial, it may be interesting to gauge experimental effects of discretizing the output variable.

In this paper, we review a recently developed rule induction method for classification. Its use for regression requires an additional data preparation step to discretize the continuous output. The final prediction involves the use of marginal votes. We compare its performance on large public domain data sets to direct approaches such as single and bagged regression trees and show that strong predictive performance can often be achieved.

2 Methods and Procedures

2.1 Regression via Classification

Although the predicted variable in regression may vary continuously, for a specific application, it's not unusual for the output to take values from a finite set, where the connection between regression and classification is stronger. The main difference is that regression values have a natural ordering, whereas for classification the class values are unordered. This affects the measurement of error. For classification, predicting the wrong class is an error no matter which class is predicted (setting aside the issue of variable misclassification costs). For regression, the error in prediction varies depending on the distance from the correct value. A central question in doing regression via classification is the following: Is it reasonable to ignore the natural ordering and treat the regression task as a classification task?

The general idea of discretizing a continuous input variable is well studied (Dougherty et al., 1995); the same rationale holds for discretizing a continuous output variable. K-means (medians) clustering (Hartigan & Wong, 1979) is simple and effective approach for clustering the output values into pseudo-classes. The values of the single output variable can be assigned to clusters in sorted order, and then reassigned by k-means to adjacent clusters. To represent each cluster by a single value, the cluster's mean value minimizes the squared error, while the median minimizes the absolute deviation.

How many classes/clusters should be generated? Depending on the application, the trend of the error of the class mean or median for a variable number of classes can be observed, and a decision made as to how many clusters are appropriate. Too few clusters would imply an easier classification problem, but put an unacceptable limit on the potential performance; too many clusters might make the classification problem too difficult. For example, Table 1 shows the global mean absolute deviation (MAD) for a typical application as the number of classes is varied. The MAD will continue to decrease with increasing number of classes and reach zero when each cluster contains homogeneous values. So one possible strategy might be to decide if the extra classes are worth the gain in terms of a lower MAD. For instance, one might decide that the extra complexity in going from 8 classes to 16 classes is not worth the small drop in MAD.

Table 1. Variation in Error with Number of Classes

Classes	1	2	4	8	16	32	64	128
MAD	4.0538	2.3532	1.2873	0.6795	0.3505	0.1784	0.0903	0.0462
SE	.0172	.0105	.0061	.0035	.0019	.0011	.0006	.0004

Figure 1 shows a simple procedure to analyze the trend using Table 1 and determine the appropriate number of classes. The basic idea is to double the number of classes, run k-means on the output variable, and stop when the reduction in the MAD from the class medians was less than a certain percentage of the MAD from using the median of all values. This percentage is adjusted by the threshold, t. In our experiments, for example, we fixed this to be 0.1 (thereby requiring can that the reduction in MAD be at least 10%). Besides the predicted variable, no other information about the data is used. If the number of unique values is very low, it is worthwhile to also try the maximum number of potential classes. In our experiments, we found that this was beneficial when there were not more than 30 unique values.

Besides helping decide the number of classes, Table 1 also provides an upper bound on performance. For example, with 16 classes, even if the classification procedure were to produce 100% accurate rules that always predicted the correct class, the use of the class median as the predicted value would imply that the regression performance could at best be 0.3505 on the training cases. This bound can be also be a factor in deciding how many classes to use.

Input: t, a user-specified threshold $(0 < t < 1)$
$Y = \{y_i, i = 1 \ldots n\}$, the set of n predicted values in the training set
Output: C, the number of classes
$M_1 :=$ mean absolute deviation (MAD) of y_i from $Median(Y)$
min-gain $:= t \cdot M_1$
$i := 1$
repeat
$C := i$
$i := 2 \cdot i$
run k-means clustering on Y for i clusters
$M_i :=$ MAD of y_i from $Median(Cluster(y_i))$
until $M_{i/2} - M_i \leq$ min-gain
output C

Fig. 1. Determining the Number of Classes

2.2 Lightweight Rule Induction

Once the regression problem is transformed into a classification task, standard classification techniques can be used. Of particular interest is a recently developed new ensemble method for learning compact disjunctive normal form (DNF) rules (Weiss & Indurkhya, 2000) that has proven to give excellent results on a wide variety of classification problems and has a time complexity that is almost linear in time relative to the number of rules and cases. This Lightweight Rule Induction (LRI) procedure is particularly interesting because it can rival the performance of very strong classification methods, such as boosted trees.

Figure 2 shows an example of a typical DNF rule generated by LRI. The complexity of a DNF rule is described with two measurements: (a) the length of a conjunctive term and (b) the number of terms (disjuncts). In this example, the rule has a length of three with two disjuncts. Complexity of rule sets generated is controlled within LRI by providing upper bounds on these two measurements.

$\{f_1 \leq 5.2$ AND $f_2 \leq 3.1$ AND $f_7 \leq .45\}$ OR
$\{f_1 \leq 2.6$ AND $f_3 \leq 3.9$ AND $f_8 \leq 5.0\} \Rightarrow$ Class1

Fig. 2. Typical DNF Rule Generated by LRI

The LRI algorithm for generating a rule for a binary classification problem is summarized in Figure 3. FN is the number of false negatives, FP is the number of false positives, and TP, the number of true positives. $e(i)$ is the cumulative number of errors for case i taken over all rules. The weighting given to a case

during induction is an integer value representing the virtual frequency of that case in the new sample. Equation 1 describes that frequency in terms of the number of cumulative errors, $e(i)$.

$$Frq(i) = 1 + e(i)^3 \tag{1}$$

Err1 is computed when TP is greater than zero. The cost of a false negative is doubled if no added condition adds a true positive. The false positives and false negatives are weighted by the relative frequency of the cases as shown in Equation 3.

$$Err1 = FP + k \cdot FN \; \{k = 1, 2, 4 ... and \; TP > 0\} \tag{2}$$

$$FP = \sum_i FP(i) \cdot frq(i); \;\; FN = \sum_i FN(i) \cdot frq(i) \tag{3}$$

1. Grow conjunctive term T until the maximum length (or until $FN = 0$) by greedily adding conditions that minimize err1.
2. Record T as the next disjunct for rule R. If less than the maximum number of disjuncts (and $FN > 0$), remove cases covered by T, and continue with step 1.
3. Evaluate the induced rule R on all training cases i and update $e(i)$, the cumulative number of errors for case i.

Fig. 3. Lightweight Rule Induction Algorithm

A detailed description of LRI and the rationale for the method are described in (Weiss & Indurkhya, 2000). Among the key features of LRI are the following:

- The procedure induces covering rules iteratively, evaluating each rule on the training cases before the next iteration, and like boosting gives more weight to erroneously classified cases in successive iterations.
- The rules are learned class by class. All the rules of a class are induced before moving to the next class. Note that each rule is actually a complete solution and contains disjunctive terms as well.
- Equal number of rules are learned for each class. All rules are of approximately the same size.
- All the satisfied rules are weighted equally, a vote is taken and the class with the most votes wins.

During the rule generation process, LRI has no knowledge about the underlying regression problem. The process is identical to that of classification. The differences come in how the case is processed after it has been classified.

2.3 Using Margins for Regression

Within the context of regression, once a case is classified, the a priori mean or median value associated with the class can be used as the predicted value. Table 2 gives a hypothetical example of how 100 votes are distributed among 4 classes. Class 2 has the most votes; the output prediction would be 2.5.

An alternative prediction can be made by averaging the votes for the most likely class with votes of classes close to the best class. In the example above, if one allows for classes with votes within 80% of the best vote to also be included, then besides the top class (class 2), class 3 need also be considered in the computation. A simple average would result in the output prediction being 2.95, and the weighted average, which we use in the experiments, gives an output prediction of 2.92.

Table 2. Voting with Margins

Class	Votes	Class-Mean
1	10	1.2
2	40	2.5
3	35	3.4
4	15	5.7

The use of margins here is analogous to nearest neighbour methods where a group of neighbours will give better results than a single neighbour. Also, this has an interpolation effect and compensates somewhat for the limits imposed by the approximation of the classes by means.

The overall regression procedure is summarized in Figure 4 for k classes, n training cases, median (or mean) value of class j, m_j, and a margin of M. The key steps are the generation of the classes, generation of rules, and using margins for predicting output values for new cases.

3 Results

To evaluate formally the performance of lightweight rule regression, several public-domain datasets were processed. The performance of our indirect approach to regression is compared to the more direct approach used in regression trees. Since our approach involves ensembles, we also compared the performance to that of bagged trees, a popular ensemble method. Because the objective is data mining, we selected datasets having relatively large numbers of cases. These were then split into train and test sets. We chose datasets from a variety of real-world applications where the regression task occurs naturally. Table 3 summarizes the data characteristics. The number of features describes numerical features and categorical variables decomposed into binary features. For each dataset, the number of unique target values in the training data is listed. Also shown is the mean

1. run k-means clustering for k clusters on the set of values $\{y_i, i = 1 \ldots n\}$
2. record the mean value m_j of the cluster c_j for $j = 1 \ldots k$
3. transform the regression data into classification data with the class label for the i-th case being the cluster number of y_i
4. apply ensemble classifier and obtain a set of rules R
5. to make a prediction for new case u, using a margin of M (where $0 \leq M \leq 1$):
   (a) apply all the rules R on the new case u
   (b) for each class i, count the number of satisfied rules (votes) v_i
   (c) class t has the most votes, v_t
   (d) consider the set of classes $P = \{p\}$ such that $v_p \geq M \cdot v_t$
   (e) the predicted output for case u, $y'_u = \frac{\sum_{j \in P} m_j v_j}{\sum_{j \in P} v_j}$

Fig. 4. Regression Using Ensemble Classifiers

Table 3. Data Characteristics

Name	Train	Test	Features	Unique Values	MAD
additive	28537	12231	10	14932	4.05
ailerons	5007	2147	40	30	3.01
ailerons2	5133	4384	6	24	1.95
census16	15948	6836	16	1819	28654
compact	5734	2458	21	54	9.54
elevator	6126	2626	18	60	.0041
kinematics	5734	2458	8	5720	.217
pole	5000	10000	48	11	29.31

absolute distance (MAD) from the median of all values. For classification, predictions must have fewer errors than simply predicting the largest class, To have meaningful results for regression, predictions must do better than the average distance from the median. This MAD is a baseline on a priori performance.

For each application, the number of classes was determined by the algorithm in Figure 1 with the user-threshold t set to 0.1. When the number of unique values was not more than 30, solutions were induced with the maximum possible classes as well.

LRI has several design parameters that affect results: (a) the number of rules per class (b) the maximum length of a rule and (c) the maximum number of disjunctions. For all of our experiments, we set the length of rules to 5 conditions. For almost all applications, increasing the number of rules increases predictive performance until a plateau is reached. In our applications, only minor changes in performance occurred after 100 rules. The critical parameter is the number of disjuncts, which depends on the complexity of the underlying concept to be learned. We varied the number of disjuncts in each rule from 1, 2, 4, 8, 16, where

1 is a rule with a single conjunctive term. The optimal number of disjuncts is obtained by validating on a portion of the training data set aside at the start.

An additional issue with maximizing performance is the use of margins. In all our experiments we included classes having vote counts within 80% of the class having the most votes.

Performance is measured in terms of error distance. The error-rates shown throughout this section are the mean absolute deviation (MAD) on test data. Equation 4 shows how this is done, where y_i and y'_i are the true and predicted values respectively for the i-th test case, and n is the number of cases in the test set.

$$MAD = \frac{1}{n}\sum_{i=1}^{n}|y_i - y'_i| \tag{4}$$

Table 4. Comparative Error for Rule sets with 10 Rules

	Number of disjuncts per rule					min-tree		sig-tree		
Name	1	2	4	8	16	error	size	error	size	SE
additive	1.85	1.81	1.72*	1.68	1.75	1.50	2620	1.52	1797	.01
ailerons	1.59	1.55*	1.63	1.66	1.75	1.46	135	1.54	514	.03
ailerons2	1.13*	1.14	1.16	1.14	1.14	1.15	51	1.23	310	.02
census16	18714	17913	17316	17457	17278*	19422	456	19604	780	412
compact	2.19	2.11	2.10*	2.10	2.14	2.25	225	2.28	514	.05
elevator	.0032*	.0034	.0034	.0038	.0036	.0025	797	.0026	332	.0001
kinematics	.162	.154	.146*	.147	.151	.154	425	.154	425	.003
pole	6.13	4.49	3.14	2.51	2.48*	2.41	578	3.57	107	.07

Table 4 summarizes the results for solutions with only 10 rules. The solutions are obtained for variable numbers of disjuncts, and the best rule solution is indicated with an asterisk. Also listed is the optimal tree solution (*min-tree*), which is the pruned tree with the minimum test error found by cost-complexity pruning. The tree size shown is the number of terminal nodes in the tree. Also shown is the tree solution where the tree is pruned by significance testing (2 sd) (*sig-tree*). The standard error is listed for the tree solution. With large numbers of test cases, almost any difference between solutions is significant. As can be seen, the simple rule-based solutions hold up quite well against the far more complex regression trees.

With greater number of rules, performance can be improved. Table 5 summarizes the results for inducing varying number of rules. All other parameters were fixed, and the number of disjuncts was determined by resampling the training cases. These results are contrasted with the solutions obtained from bagging regression trees (Breiman, 1996). 500 trees were used in the bagged solutions. Note that the complexity of the bagged solutions is very high – the individual trees are unpruned trees which, for regression problems, are extremely large.

Table 5. Comparative Error for Different Number of Rules

	Number of rules in LRI-solution						
Name	25	50	100	250	500	Bagged Trees	SE
additive	1.40	1.32	1.30	1.28	1.27	1.00	.01
ailerons	1.44	1.43	1.41	1.38	1.39	1.19	.02
ailerons2	1.10	1.10	1.10	1.11	1.10	1.10	.01
census16	15552	15093	14865	14537	14583	16008	258
compact	1.91	1.84	1.82	1.80	1.80	1.68	.02
elevator	.0030	.0030	.0030	.0030	.0030	.0036	.0001
kinematics	.128	.124	.121	.119	.120	.112	.001
pole	2.09	1.97	1.96	1.96	1.96	2.32	.05

The number of classes is specified prior to rule induction, and it affects the complexity of solutions. Varying this parameter gives the typical performance versus complexity trend – improved performance with increasing complexity until the right complexity fit is reached, and then decreased performance with further increases in complexity.

4 Discussion

Lightweight Rule Induction has an elementary representation for classification. Scoring is trivial to understand: the class with the most satisfied rules wins. To perform regression, the output variable is discretized, and all rules for a single class are associated with a single discrete value. Clearly, this is a highly restrictive representation, reducing the space of continuous outputs to a small number of discrete values, and the space of values of rules and disjuncts to a single value per class.

The key question is whether a simple transformation from regression to classification can retain high quality results. In Section 3, we presented results from public domain datasets that demonstrate that our approach can indeed produce high quality results.

For best predictive performance, a number of parameters must be selected prior to running. We have concentrated on data mining applications where it can be expected that sufficient test sets are available for parameter estimation. Thus, we have included results that describe the minimum test error. With big data, it is easy to obtain more than one test sample, and for estimating a single variable, a large single test set is adequate in practice (Breiman et al., 1984). For purposes of experimentation, we fixed almost all parameters, except for maximum number of disjuncts and the number of rules. The number of disjuncts is clearly on the critical path to higher performance. Its best value can readily be determined by resampling on the large number of training cases.

The use of k-means clustering to discretize the output variable, producing pseudo-classes, creates another task for estimation. What is the proper number of classes? The experimental results suggest that when the number of unique

values is modest, perhaps 30 or less, then using that number of classes is feasible and can be effective. For true continuous output, we used a simple procedure for analyzing the trend as the number of classes is doubled. This type of estimate is generally quite reasonable and trivially obtainable, but occasionally, slightly more accurate estimates can be found by trying different numbers of classes, inducing rules, and testing on independent test data.

A class representation is an approximation that has a potential sources of error beyond those found for other regression models. For a given number of classes less than the number of unique values, the segmentation error, measured by the MAD of the median values of the classes, is a lower bound on predictive performance. For pure classification, where the most likely class is selected, the best that the method can do is the MAD for the class medians. In the experimental results, we see this limit for the artificial *additive* data generated from an exact function (with additive random noise). With a moderate number of classes, the method is limited by this approximation error. To reduce the minimum error implied by the class medians, more classes are needed. That in turn leads to a much more difficult classification problem, also limiting predictive performance.

Minimizing classification error is not the same as minimizing deviation from the true value. This difference introduces another type of approximation error in our regression process. This error is most obvious when we predict using the median value of the single most likely class. We have presented an alternative that capitalizes on a voting method's capability to identify close votes. Thus by averaging the values of the most likely class and its close competitors, as determined by the number of votes, more accurate results are achieved. The analogy is to nearest-neighbor methods, where with large samples, a group of neighbors will perform better than the single best neighbor. Averaging the neighbors also has an interpolation effect, somewhat counterbalancing the implicit loss of accuracy of using the median approximation to a class value.

Overall, the lightweight rule regression methods presented here are straightforward to implement. Where error is measured relative to the median value of all examples, LRI often widely surpassed tree regression and rivaled the bagged tree results. Depending on the number of rules induced, the rule based solution can be remarkably simple in presentation. Although there are a number of parameters that must be estimated, effective solutions can be achieved by judicious use of test data or by a priori knowledge of user preferences.

References

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting and variants. *Machine Learning*, *36*, 105–139.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, *24*, 123–140.

Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Monterrey, CA.: Wadsworth.

Cohen, W., & Singer, Y. (1999). A simple, fast, and effective rule learner. *Proceedings of Annual Conference of American Association for Artificial Intelligence* (pp. 335–342).

Dougherty, J., Kohavi, R., & Sahami, M. (1995). Supervised and unsupervised discretization of continuous features. *Proceedings of the 12th Int'l Conference on Machine Learning* (pp. 194–202).

Friedman, J., Hastie, T., & Tibshirani, R. (1998). *Additive logistic regression: A statistical view of boosting* (Technical Report). Stanford University Statistics Department. www.stat-stanford.edu/∼tibs.

Hartigan, J., & Wong, M. (1979). A k-means clustering algorithm, ALGORITHM AS 136. *Applied Statistics*, *28*.

Schapire, R., Freund, Y., Bartlett, P., & Lee, W. (1998). Boosting the margin: A new explanation for the effectiveness of voting methods. *Proceedings of the Fourteenth Int'l Conference on Machine Learning* (pp. 322–330). Morgan Kaufmann.

Weiss, S., & Indurkhya, N. (1995). Rule-based machine learning methods for functional prediction. *Journal of Artificial Intelligence Research*, *3*, 383–403.

Weiss, S., & Indurkhya, N. (2000). Lightweight rule induction. *Proceedings of the Seventeenth International Conference on Machine Learning* (pp. 1135–1142).

Learning XML Grammars

Henning Fernau

Wilhelm-Schickard-Institut für Informatik,
Universität Tübingen,
Sand 13, D-72076 Tübingen, Germany
fernau@informatik.uni-tuebingen.de

Abstract. We sketch possible applications of grammatical inference techniques to problems arising in the context of XML. The idea is to infer document type definitions (DTDs) of XML documents in situations either when the original DTD is missing or when a DTD should be (re)designed or when a DTD should be restricted to a more user-oriented view on a subset of the (given) DTD. The usefulness of such an approach is underlined by the importance of knowing appropriate DTDs; this knowledge can be exploited, e.g., for optimizing database queries based on XML.

1 Introduction

This paper exhibits possible applications of Machine Learning techniques, especially, of grammatical inference, within the area of XML technologies, or, more precisely, of syntactical aspects of XML formalized by XML grammars. In the introduction, firstly we briefly comment on XML and Machine Learning in general, then we sketch application scenarios of grammatical inference within XML practice, and finally we show the paper's value for the grammatical inference community.

XML. The expectations surrounding XML (eXtendible Markup Language) as a universal syntax for data representation and exchange on the world wide web continues to grow. This is underlined by the amount of effort being committed to XML by the World Wide Web Consortium (W3C) (see `www.w3.org/TR/REC-XML`), by the huge number of academics involved in the research of the backgrounds of XML, as well as by numerous private companies. Moreover, an ever-growing number of applications arise which make use of XML, although they are not directly related to the world wide web. For example, nowadays XML plays an important role in the integration of manufacturing and management in highly automated fabrication processes such as in car companies [12]. Further information on XML can be found under `www.oasis-open.org/cover/xmlIntro.html`.

XML grammars. The syntactic part of the XML language describes the relative position of pairs of corresponding *tags*. This description is done by means of a *document type definition* (DTD). Ignoring attributes of tags, a DTD is a special form of a context-free grammar. This sort of grammar formalism has been

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 73–87, 2001.

formalized and studied by Berstel and Boasson [8] by what they termed *XML grammars.*[1]

Machine learning is nowadays an active research area with a rather diverse community ranging from practitioners in industry to pure mathematicians in academia. While, generally speaking, stochastic approaches are prominent as machine learning techniques in many applied areas like pattern recognition in order to capture noise phenomena, there are also application domains—like the inference of document type definitions presented in this paper—where a deterministic learning model is appropriate. The sub-area of Machine Learning which deals with the inference of grammars, automata or other language describing devices is called grammatical inference. This will be the part of Machine Learning we deal with in this paper. For other aspects of Machine Learning, we refer to [9,28,30]. For a general treatment of the use of machine learning within data mining, see Mitchell [29]. Especially promising in this respect seem to be combinations of syntactical and statistical approaches, see Freitag [19].

Grammatical inference. Our paper can also be seen as a contribution to further promote the use of machine learning techniques within database technologies, in particular, when these are based on the XML framework. More specifically, we discuss learnability issues for XML grammars. This question is interesting for several reasons:

Three applications of grammatical inference. As already worked out by Ahonen, grammatical inference (GI) techniques can be very useful for automatic document processing, see [2,3]. More specifically, Ahonen detailed on the following two applications of the inference of DTDs (of HTML documents) [1,2]:

Firstly, GI techniques can be used to assist designing grammars for (semi-) structured documents. This is often desirable, since either the system users are not experts in grammar design or the grammars are rather huge and difficult to handle. The user feeds several examples of syntactically correct tagged documents into the GI system, which then suggests a grammar describing these documents. In this application, an interaction between the human grammar designer and the GI system is desirable, e.g., for coping with erroneous examples, or when previous grammar design decisions are modified. If the given examples are not originally tagged (e.g., if they do not stem from an XML document), document recognition techniques can be applied in a first step, see [4,25,33]. Fankhauser and Xu integrate both steps in their system [14].

Secondly, GI may be of help in creating views and subdocuments. For several applications, standard DTDs have been proposed. However, these DTDs are usually large and designed to cover many different needs. GI may be used to find reasonable smaller subsets of the corresponding document descriptions.

Note that Ahonen used a rather direct approach to the inference of DTDs by simply inferring right-hand sides of rules (as regular sets). Unfortunately, in

[1] Also, Behrens and Buntrock [7] investigated formal language properties of DTDs.

this way grammars might be derived which do not satisfy the requirements of an XML grammar. Therefore, our approach is necessary and more adequate for XML documents.

We mention a third application of the inference of DTDs for XML documents in connection with databases: The importance of making use of DTDs —whenever known— to optimize the performance of database queries based on XML has been stressed by various authors, see [35] and the literature quoted therein. Unfortunately, DTDs are not always transferred when XML documents are transmitted. Therefore, an automatic generation of DTDs can be useful in this case, as well.

A contribution to the GI community. Finally, one can consider this paper also as a contribution to the GI community: Many GI results are known for regular languages, but it seems to be hard to get beyond. This has been formulated as a challenge by de la Higuera in a recent survey article [24]. Many authors try to transfer learnability results from the regular language case to the nonregular case by preprocessing. Some of these techniques are surveyed in [18]. Here, we develop a similar preprocessing technique for XML grammars, focussing on a learning model known as *identification in the limit from positive samples* or *exact learning from text.*

Summary of the paper. The paper is structured as follows. In Section 2, we present XML grammars as introduced by Berstel and Boasson. Section 3 reviews the necessary concepts from the algorithmics of identifying regular languages. In Section 4, we show how to apply the results of Section 3 to the automatic generation of DTDs for XML documents. Finally, we summarize our findings and outline further aspects and prospects of GI issues in relation with XML.

2 XML Grammars

In this section, we will present the framework of XML grammars exhibited by Berstel and Boasson and relate them to regular languages. This will be the key for obtaining learning algorithms for XML grammars.

Definition and Examples. Berstel and Boasson gave the following formalization of an XML grammar:

Definition 1. *An* XML grammar *is composed of a terminal alphabet* $T = A \cup \bar{A}$ *with* $\bar{A} = \{\bar{a} \mid a \in A\}$*, of a set of variables* $V = \{X_a \mid a \in A\}$*, of a distinguished variable called the* axiom *and, for each letter* $a \in A$*, of a regular set* $R_a \subseteq V^*$ *which defines the (possibly infinite) set of productions* $X_a \to am\bar{a}$ *with* $m \in R_a$ *and* $a \in A$*. We also write* $X_a \to aR_a\bar{a}$ *for short.*

An XML language *is generated by some XML grammar.*

Note that the syntax of document type definitions (DTDs) as used in XML differs, at first glance, from the formalization of Berstel and Boasson, but the transfer is done easily.

Example 1. For example, the (rather abstract) DTD

$$\begin{array}{l} <!\texttt{DOCTYPE}\ a\ [\\ \qquad <!\texttt{ELEMENT}\ a \quad ((a|b),(a|b))> \\ \qquad <!\texttt{ELEMENT}\ b \quad (b)*\ > \\ ]> \end{array}$$

would be written as:

$$\begin{aligned} X_a &\to a(X_a|X_b)(X_a|X_b)\bar{a} \\ X_b &\to b(X_b)^*\bar{b} \end{aligned}$$

with axiom X_a.

Interpreting b as open bracket and $\bar{b}$ as close bracket, it is easy to see that the words derivable from X_b correspond to all the syntactically correct bracketizations. For example, $w = bb\bar{b}bb\bar{b}\bar{b}$ is derived as

$$X_b \Rightarrow bX_bX_b\bar{b} \Rightarrow bb\bar{b}X_b\bar{b} \Rightarrow bb\bar{b}X_b\bar{b}\bar{b} \Rightarrow w.$$

This issue is furthered in Example 2.

In other words, an XML grammar corresponds to a DTD in a natural fashion and vice versa. As to the syntax of DTDs, the axiom of the grammar is introduced by `DOCTYPE`, and the set of rules associated to a tag by `ELEMENT`. Indeed, an element is composed of a *type* and a *content model*. Here, the type is the tag name and the content model is a regular expression for the right-hand sides of the rules for this tag. We finally remark that *entities* as well as `#PCDATA` (i.e., textual) information are ignored in the definition of XML grammars. Below, we will show that it is easy to cope with the textual information.

Example 2. Let $A = \{a_1, \ldots, a_n\}$. The language D_A of *Dyck primes* over $T = A \cup \bar{A}$, generated by

$$\begin{aligned} X &\to X_{a_1}|\ldots|X_{a_n}, \text{ where, for } a \in A, \\ X_a &\to a(X_{a_1}|\ldots|X_{a_n})^*\bar{a} \end{aligned}$$

with axiom X is not an XML language. However, each variable X_{a_i} of this grammar generates the XML language

$$D_{a_i} := D_A \cap a_i(A \cup \bar{A})^*\bar{a}_i.$$

In particular, $D_{\{a_1\}}$ is an XML language.

Simple properties. By definition of an XML grammar, the following is quite clear:

Lemma 1. *If $L \subseteq (A \cup \bar{A})^*$ is an XML language, then $L \subseteq D_A$.*

Therefore, Berstel and Boasson derived necessary and sufficient conditions for a subset L of D_A to be an XML language.

We now give some notions we need for stating some of these conditions. We denote by $F(L)$ the set of *factors* of $L \subseteq \Sigma^*$, i.e., $F(L) = \{x, y, z \in \Sigma^* \mid xyz \in L\}$. For $L \subseteq (A \cup \bar{A})^*$, let $F_a(L) = D_a \cap F(L)$ be the set of those factors in L that are also Dyck primes starting with letter $a \in A$. Using these notions, we may sharpen the previous lemma as follows:

Lemma 2. *If $L \subseteq (A \cup \bar{A})^*$ is an XML language, then $L = F_a(L)$ for some $a \in A$.*

Characterizing XML languages via regular languages. Consider $w \in D_a$. w is uniquely decomposable as $w = au_{a_1}u_{a_2}\dots u_{a_n}\bar{a}$, with $u_{a_i} \in D_{a_i}$ for $i = 1, \dots, n$. The *trace* of w is defined as $a_1 \dots a_n \in A^*$. The set $S_a(L)$ of all traces of words in $F_a(L)$ is called the *surface of $a \in A$ in $L \subseteq D_A$*.

Surfaces are useful for defining XML grammars. Consider a family $\mathcal{S} = \{S_a \mid a \in A\}$ of regular languages over A. The *standard XML grammar* $G_{\mathcal{S}}$ associated to $\mathcal{S}$ is defined as follows. The set of variables is $V = \{X_a \mid a \in A\}$. For each $a \in A$, let $R_a = \{X_{a_1} \dots X_{a_n} \mid a_1 \dots a_n \in S_a\}$ and consider the rules $X_a \to aR_a\bar{a}$. By definition, $G_{\mathcal{S}}$ is indeed an XML grammar for any choice of the axiom. Moreover, for each language L_a generated from axiom X_a by using the rules of $G_{\mathcal{S}}$, it can be shown that $S_a(L_a) = S_a$.

Now, consider for a family $\mathcal{S} = \{S_a \mid a \in A\}$ of regular languages over A and some fixed letter $a_0 \in A$ the family $\mathcal{L}(\mathcal{S}, a_0)$ of those languages languages $L \subseteq D_{a_0}$ such that $S_a(L) = S_a$ for all $a \in A$. Since $\mathcal{L}(\mathcal{S}, a_0)$ is closed under (arbitrary) union, there is a maximal element in this family. Berstel and Boasson derived the following nice characterization [8, Theorem 4.1]:

Theorem 1. *Consider a family $\mathcal{S} = \{S_a \mid a \in A\}$ of regular languages over A and some fixed letter $a_0 \in A$. The language generated by the standard XML grammar $G_{\mathcal{S}}$ with axiom X_{a_0} is the maximal element of the family $\mathcal{L}(\mathcal{S}, a_0)$. Moreover, this is the only XML language in $\mathcal{L}(\mathcal{S}, a_0)$.*

Finally, [8, Proposition 3.8] yields:

Lemma 3. *If L is an XML language, then there exists a standard XML grammar generating L.*

Therefore, there is a *one-to-one correspondence between surfaces and XML languages*. This is the key observation for transferring learnability results known for regular languages to XML languages.

3 A Learning Scenario

Gold-style learning. When keeping in mind the possible applications of inferring XML grammars, the typical situation is that an algorithm is needed that, given a set of examples that should fit the sought DTD, proposes a valid DTD. This

corresponds to the learning model *identification in the limit from positive samples*, also known as *exact learning from text*, which was introduced by Gold [22] and has been studied thoroughly by various authors within the computational learning theory and the grammatical inference communities.

Definition 2. *Consider a language class $\mathcal{L}$ defined via a class of language describing devices $\mathcal{D}$ as, e.g., grammars or automata. $\mathcal{L}$ is said to be* identifiable *if there is a so-called* inference machine *IM to which as input an arbitrary language $L \in \mathcal{L}$ may be enumerated (possibly with repetitions) in an arbitrary order, i.e., IM receives an infinite input stream of words $E(1), E(2), \ldots$, where $E : \mathbf{N} \to L$ is an enumeration of L, i.e., a surjection, and IM reacts with an output stream $D_i \in \mathcal{D}$ of devices such that there is an $N(E)$ so that, for all $n \geq N(E)$, we have $D_n = D_{N(E)}$ and, moreover, the language defined by $D_{N(E)}$ equals L.*

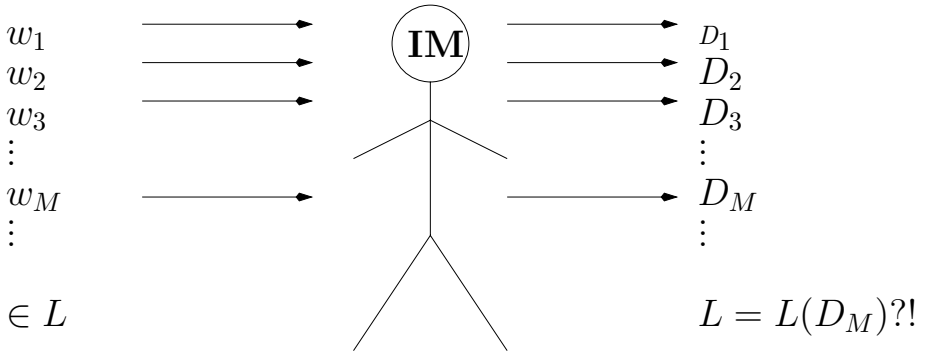


Fig. 1. Gold's learning scenario

Figure 1 tries to illustrate this learning scenario for a fixed language class $\mathcal{L}$ described by the device class $\mathcal{D}$. Often, it is convenient to view IM mapping a finite sample set $I_+ = \{w_1, \ldots, w_M\}$ to a hypothesis D_M. The aim is then to find algorithms which, given I_+, produce a hypothesis D_M describing a language $L_M \supseteq I_+$ such that, for any language $L \in \mathcal{L}$ which contains I_+, $L_M \subseteq L$. In other words, L_M is the smallest language in $\mathcal{L}$ extending I_+.

Gold [22] has already established:

Lemma 4. *The class of regular languages is not identifiable.*

This result readily transfers to XML languages:

Lemma 5. *The class of all XML languages (over a fixed alphabet) is not identifiable.*

Identifiable regular subclasses. Since we think that the inference of XML grammars has important practical applications (as detailed in the Introduction), we

show how to define identifiable subclasses of the XML languages. To this end, we reconsider the identification of subclasses of the regular languages, because XML grammars and regular languages are closely linked due to the one-to-one correspondence of XML standard grammars and regular surfaces as stated in the preceding section.

Since the regular languages are a very basic class of languages, many attempts have been made to find useful identifiable subclasses of the regular languages. According to Gregor [23], among the most popular identifiable regular language classes are the k-reversible languages [5] and the terminal-distinguishable languages [31,32]. Other identifiable subclasses are surveyed in [27]. A nice overview of the involved automata and algorithmic techniques can be found in [13]. Recently, we developed a framework which generalizes the explicitly mentioned language classes in a uniform manner [15]. We will briefly introduce this framework now.

Definition 3. *Let F be some finite set. A mapping $f : T^* \to F$ is called a* distinguishing function *if $f(w) = f(z)$ implies $f(wu) = f(zu)$ for all $u, w, z \in T^*$.*

$L \subseteq T^*$ is called f-distinguishable *if, for all $u, v, w, z \in T^*$ with $f(w) = f(z)$, we have $zu \in L \iff zv \in L$ whenever $\{wu, wv\} \subseteq L$.*

The family of f-distinguishable languages (over the alphabet T) is denoted by (f, T)-DL.

For $k \geq 0$, the example $f(x) = \sigma_k(x)$ (where $\sigma_k(x)$ is the suffix of length k of x if $|x| \geq k$, and $\sigma_k(x) = x$ if $|x| < k$) leads to the k-reversible languages, and $f(x) = \mathrm{Ter}(x) = \{\, a \in T \mid \exists u, v \in T^* : uav = x \,\}$ yields (reversals of) the terminal-distinguishable languages.

We derived another characterization of (f, T)-DL based on automata [15].

Definition 4. *Let $\mathcal{A} = (Q, T, \delta, q_0, Q_F)$ be a finite automaton. Let $f : T^* \to F$ be a distinguishing function. A is called* f-distinguishable *if:*

1. *$\mathcal{A}$ is deterministic.*
2. *For all states $q \in Q$ and all $x, y \in T^*$ with $\delta^*(q_0, x) = \delta^*(q_0, y) = q$, we have $f(x) = f(y)$.*
   (In other words, for $q \in Q$, $f(q) := f(x)$ for some x with $\delta^*(q_0, x) = q$ is well-defined.)
3. *For all $q_1, q_2 \in Q$, $q_1 \neq q_2$, with either (i) $q_1, q_2 \in Q_F$ or (ii) there exist $q_3 \in Q$ and $a \in T$ with $\delta(q_1, a) = \delta(q_2, a) = q_3$, we have $f(q_1) \neq f(q_2)$.*

Intuitively speaking, a distinguishing function f can be seen as an oracle which can be used in order to resolve possible backward nondeterminisms within, e.g., the minimal deterministic finite automaton accepting a language $L \in (f, T)$-DL. For example, the automaton $\mathcal{A} = (\{1, 2\}, \{a, b\}, \delta, 1, \{2\})$ with $\delta(1, a) = \delta(2, a) = \delta(2, b) = 2$ accepts $L = \{a\}\{a, b\}^*$. $\mathcal{A}$ is not σ_0-distinguishable (i.e, not 0-reversible in the dictum of Angluin [5]), since condition 3.(ii) in the above definition is violated: choose $q_1 = 1$ and $q_2 = 2$ and $q_3 = 2$ with $\delta(q_1, a) = \delta(q_2, a) = q_3$. Also, $\mathcal{A}$ is not Ter-distinguishable, since condition 2. is violated: both a and ab lead from the start state to state 2, but $\mathrm{Ter}(a) \neq \mathrm{Ter}(ab)$.

Nevertheless, $L \in (\mathrm{Ter}, \{a,b\})$-DL, since the automaton $\mathcal{A}' = (\{p,q,r\}, \{a,b\}, \delta', p, \{q,r\})$ with $\delta'(p,a) = \delta'(q,a) = q$ and $\delta'(q,b) = \delta'(r,a) = \delta'(r,b) = r$ also accepts L. Moreover, $\mathrm{Ter}(p) = \emptyset$, $\mathrm{Ter}(q) = \{a\}$ and $\mathrm{Ter}(r) = \{a,b\}$ are well-defined.

Theorem 2. *A language is f-distinguishable iff it is accepted by an f-distinguishable automaton.*

We return now to the issue of learning. In [15], we have shown the following theorem:

Theorem 3. *For each alphabet T and each distinguishing function $f : T \to F$, the class (f,T)-DL is identifiable.*

Moreover, there is an identification algorithm which, given the finite sample set $I_+ \subset T^*$, yields a finite automaton hypothesis $\mathcal{A}$ in time $O(\alpha(|F|n)|F|n)$, where α is the inverse Ackermann function[2] and n is the total length of all words in I_+.

The language recognized by $\mathcal{A}$ is the smallest f-distinguishable language containing I_+.

Remark 1. Since, in principle, the language classes (f,T)-DL grow when the size of the range F of f grows, the algorithm mentioned in the preceding theorem offers a natural trade-off between precision (i.e., getting more and more of the regular languages) and efficiency. From another viewpoint, f can be seen as the *explicit bias* or *commitment* one has to make when learning regular languages from text exactly. Since, due to Lemma 4, restricting the class of regular languages towards identifiable subclasses cannot be circumvented, having an explicit and well-formalized bias which characterizes the identifiable language class is of natural interest.

A merging state inference algorithm. For reasons of space, we will only sketch the inference algorithm. Note that the algorithm is a merging state algorithm similar to the algorithm for inferring 0-reversible languages as developed by Angluin [5].

Consider an input sample set $I_+ = \{w_1, \ldots, w_M\} \subseteq T^+$ of the inference algorithm. Let $w_i = a_{i1} \ldots a_{in_i}$, where $a_{ij} \in T$, $1 \leq i \leq M$, $1 \leq j \leq n_i$. We are going to describe a simple nondeterministic automaton accepting exactly I_+. Namely, the *skeletal automaton* for the sample set is defined as

$$\begin{aligned} A_S(I_+) &= (Q_S, T, \delta_S, Q_0, Q_f), \quad \text{where} \\ Q_S &= \{\, q_{ij} \mid 1 \leq i \leq M, 1 \leq j \leq n_i + 1 \,\}, \\ \delta_S &= \{\, (q_{ij}, a_{i,j+1}, q_{i,j+1}) \mid 1 \leq i \leq M, 1 \leq j \leq n_i \,\}, \\ Q_0 &= \{\, q_{i1} \mid 1 \leq i \leq M \,\} \quad \text{and} \\ Q_f &= \{\, q_{i,n_i+1} \mid 1 \leq i \leq M \,\}. \end{aligned}$$

Observe that we allow a *set* of initial states. The *frontier string* of q_{ij} is defined by $\mathrm{FS}(q_{ij}) = a_{ij} \ldots a_{in_i}$. The *head string* of q_{ij} is defined by $\mathrm{HS}(q_{ij}) = a_{i1} \ldots a_{i,j-1}$.

[2] as defined by Tarjan [34]; α is an extremely slowly growing function

In other words, $\mathrm{HS}(q_{ij})$ is the unique string leading from an initial state into q_{ij}, and $\mathrm{FS}(q_{ij})$ is the unique string leading from q_{ij} into a final state. Therefore, the skeletal automaton of a sample set simply spells all words of the sample set in a trivial fashion. Since there is only one word leading to any q, namely $\mathrm{HS}(q)$, $f(q) = f(\mathrm{HS}(q))$ is well-defined.

Now, for $q_{ij}, q_{k\ell} \in Q_S$, define $q_{ij} \rightleftharpoons_f q_{k\ell}$ iff (1) $\mathrm{HS}(q_{ij}) = \mathrm{HS}(q_{k\ell})$ or (2) $\mathrm{FS}(q_{ij}) = \mathrm{FS}(q_{k\ell})$ as well as $f(q_{ij}) = f(q_{k\ell})$. In general, $\rightleftharpoons_f$ is not an equivalence relation. Hence, define $\equiv_f := (\rightleftharpoons_f)^+$, denoting in this way the transitive closure of the original relation. Then, we can prove:

Lemma 6. *For each distinguishing function f and each sample set I_+, $\equiv_f$ is an equivalence relation on the state set of $\mathcal{A}_S(I_+)$.*

The gist of the inference algorithm is to merge $\equiv_f$-equivalent states of $\mathcal{A}_S(I_+)$. Formally speaking, the notion of quotient automaton construction is needed. We briefly recall this notion:

A *partition* of a set S is a collection of pairwise disjoint nonempty subsets of S whose union is S. If π is a partition of S, then, for any element $s \in S$, there is a unique element of π containing s, which we denote as $B(s, \pi)$ and call the *block* of π containing s. A partition π is said to *refine* another partition π' iff every block of π' is a union of blocks of π. If π is any partition of the state set Q of the automaton $\mathcal{A} = (Q, T, \delta, q_0, Q_F)$, then the *quotient automaton* $\pi^{-1}\mathcal{A} = (\pi^{-1}Q, T, \delta', B(q_0, \pi), \pi^{-1}Q_F)$ is given by $\pi^{-1}\hat{Q} = \{\, B(q, \pi) \mid q \in \hat{Q} \,\}$ (for $\hat{Q} \subseteq Q$) and $(B_1, a, B_2) \in \delta'$ iff $\exists q_1 \in B_1 \exists q_2 \in B_2 : (q_1, a, q_2) \in \delta$.

We consider now the automaton $\pi_f^{-1}\mathcal{A}_S(I_+)$, where π_f is the partition induced by the equivalence relation $\equiv_f$. We have shown [17]:

Theorem 4. *For each distinguishing function f and each sample set I_+, the automaton $\pi_f^{-1}\mathcal{A}_S(I_+)$ is an f-distinguishable automaton.*

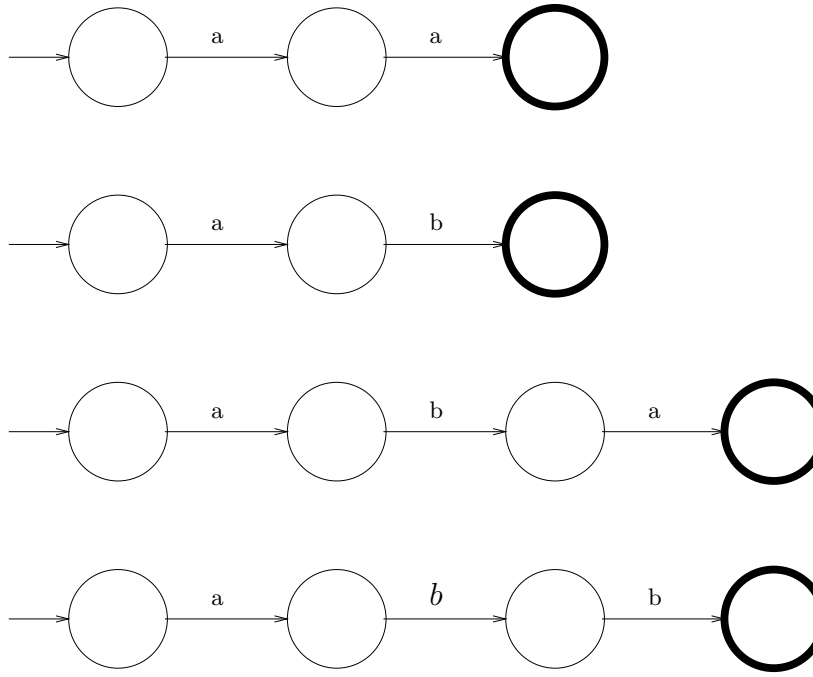
Moreover, the language accepted by $\pi_f^{-1}\mathcal{A}_S(I_+)$ is the smallest f-distinguishable language containing I_+.

Therefore, it suffices to compute $\mathcal{A}_S(I_+)$, $\equiv_f$ and, finally, $\pi_f^{-1}\mathcal{A}_S(I_+)$, in order to obtain a correct hypothesis in the sense of Gold's model. Observe that the notion of quotient automaton formalizes the intuitive idea of "merging equivalent states."

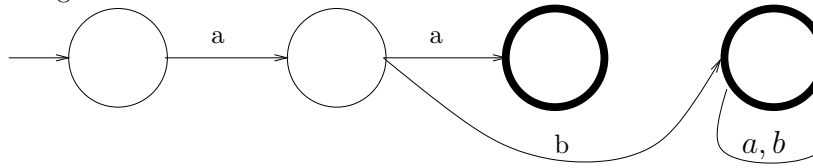
If the intended language is enumerated "completely" as indicated in Figure 1, then an on-line version of the above-sketched procedure is of course preferable. Such an algorithm is obtainable as in the case of reversible languages [5].

We conclude this section with two abstract examples. As to the inference of DTDs, we refer to the next section.

Example 3. We consider the distinguishing function $f = \mathrm{Ter}$ and the sample $I_+ = \{aa, ab, aba, abb\}$. The skeletal automaton looks as follows:



Due to the first merging condition alone (i.e., equal head strings), the merging state algorithm would produce a so-called prefix-tree acceptor for I_+ (which is usually employed as the starting place for other merging state algorithms like Angluin's [5], also see [15] for function distinguishable languages); for example, all four initial states are merged into one intial state and all four states reachable by a-transitions from an initial state are merged into another state. Due to the second merging condition (i.e., equal frontier strings and equal Ter-values of the head strings), the "last" three final states are merged in addition. This way, the following automaton results:



Example 4. We consider the distinguishing function $f = \sigma_0$ and the sample $I_+ = \{aatp, atp\}$. In the corresponding skeletal automaton $\mathcal{A}_S(I_+)$, $q_{11} \equiv_{\sigma_0} q_{21}$, since $\mathrm{HS}(q_{11}) = \mathrm{HS}(q_{21}) = \lambda$ and $q_{12} \equiv_{\sigma_0} q_{22}$, since $\mathrm{HS}(q_{12}) = \mathrm{HS}(q_{22}) = a$; similarly, $q_{15} \equiv_{\sigma_0} q_{26}$, $q_{14} \equiv_{\sigma_0} q_{25}$ $q_{13} \equiv_{\sigma_0} q_{24}$, and $q_{12} \equiv_{\sigma_0} q_{23}$, since the corresponding frontier strings are the same. Therefore, an automaton accepting a^+tp will result.

4 Learning Document Type Definitions

An XML grammar identification algorithm. We propose the following strategy for inferring XML grammars.

Algorithm 1 *(Sketch)*

1. *Firstly, one has to commit oneself to a distinguishing function f formalizing the bias of the learning algorithm.*
2. *Then, the sample XML document has to be transformed into sets of positive samples, one such sample set I_+^a for each surface which has to be learned.*
3. *Thirdly, each I_+^a is input to an identification algorithm for f-distinguishable languages, yielding a family $\mathcal{S} = \{S_a \mid a \in A\}$ of regular f-distinguishable languages over A.*
4. *Finally, the corresponding XML standard grammar is output.*

Remark 2. Let us comment on the first step of the sketched algorithm. Due to Lemma 5, it is impossible to identify any XML language in the limit from positive samples. Note 1 explains the advantage of having an explicit bias in such situations. Choosing a bias can be done in an incremental manner, starting with the trivial distinguishing function which characterizes the 0-reversible languages and integrating more and more features into the chosen distinguishing function whenever appropriate. This is also important due to the exponential dependence of the running time of the employed algorithm on the size of the range of the chosen distinguishing function, see Theorem 3. Conversely, a too simplistic commitment would entail the danger of "over-generalization" which is a frequently discussed topic in GI. Hence, when a user encounters a situation where the chosen algorithm generalizes too early or too much, she may choose a more sophisticated distinguishing function.

Remark 3. Of course, it is also possible to use identifiable language classes different from the f-distinguishable languages in order to define identifiable subclasses of XML languages. For example, Ahonen [2,3] proposed taking a variant of what is known as k-testable languages [20] (which is basically a formalization of the empiric k-gram approach well-known in pattern recognition, see the discussion in [16]).

Remark 4. Theorem 4 immediately implies that the class $\mathrm{XML}(f, A)$ of XML languages over the tag alphabet $T = A \cup \bar{A}$ whose surface is f-distinguishable is identifiable by means of Algorithm 1.

A bookstore example. Let us clarify the procedure sketched in Algorithm 1 by an extended example:

Example 5. We discuss a bookstore which would like to prepare its internet appearance by transforming its offers into XML format. Consider the following entry for a book:

```
<book>
   <author><last-name>Abiteboul</last-name></author>
   <author><last-name>Vercoustre</last-name></author>
   <title>Research and Advanced Technology for Digital Libraries.
       Third European Conference</title>
   <price>56.24 Euros</price>
</book>
```

Further, assume that for $f : \Sigma \to F$, $|F| = 1$, i.e., we are considering the distinguishing function f corresponding to the 0-reversible languages in the dictum of Angluin [5]. First, let us rewrite the given example in the formalism of Berstel and Boasson. To this end, let X_b correspond to the tag pair `<book>` and `</book>`, X_a correspond to `<author>` and `</author>`, X_n correspond to `<last-name>` and `</last-name>`, X_t correspond to `<title>` and `</title>`, and X_p correspond to `<price>` and `</price>`. Let us further write each tag pair belonging to variable X_y as $y, \bar{y}$ as in the examples above. The given concrete book example then reads as $w = ban\bar{n}\bar{a}an\bar{n}\bar{a}t\bar{t}p\bar{p}\bar{b}$. Here, we ignore an arbitrary data text. Obviously, $w \in D_b$. We find the decomposition $w = bu_au_au_tu_p\bar{b}$, with $u_a = an\bar{n}\bar{a} \in D_a$, $u_t = t\bar{t} \in D_t$ and $u_p = p\bar{p} \in D_p$. The trace belonging to w is, therefore, $aatp$. By definition, $aatp$ belongs to the surface S_b which has to be learned.

Consider as a second input example:

```
<book>
   <author><last-name>Thalheim</last-name></author>
   <title>Entity-Relationship Modeling.
       Foundations of Database Technology</title>
   <price>50.10 Euros</price>
</book>
```

From this example, we may infer that atp belongs to S_b, as well. The inference algorithm for 0-reversible languages would now yield the hypothesis $S_b = a^+tp$ (see Example 4), which is, in fact, a reasonable generalization for our purpose, since a book in a bookstore will probably be always specified by a non-empty list of authors, its title and its price. Incorporating arbitrary data text (`#PCDATA`) by means of a place-holder τ in a natural fashion, the following XML grammar will be inferred:

$$\begin{aligned} X_b &\to bR_b\bar{b} \text{ with } R_b = \{X_a^j X_t X_p \mid j > 0\}, \\ X_a &\to aR_a\bar{a} \text{ with } R_a = \{X_n\}, \\ X_n &\to n\tau\bar{n}, \\ X_t &\to t\tau\bar{t}, \\ X_p &\to p\tau\bar{p}. \end{aligned}$$

We conclude this section with a remark concerning a special application described in the introduction.

Remark 5. When creating restricted or specialized views on documents (which is one of the possible inference tasks proposed by Ahonen), one can assume that the large DTD is known to the inference algorithm. Then, it is, of course,

useless to infer regular languages which are not subsets of the already given "maximal" surfaces S_a. Therefore, it is reasonable to take as "new" hypothesis surfaces $S'_a \cap S_a$, where S'_a is the surface output by the employed regular language inference algorithm.

5 Conclusions

Our findings. We presented a method which allows us to transfer results known from the learning of regular languages towards the learning of XML languages. We will provide a competitive implementation of our algorithms shortly via the WWW.

Two further applications. The derivation of DTDs is not the only possible application of GI techniques in XML design. Another important issue is the design of appropriate *contexts*. For example, Brüggemann-Klein and Wood [10,11] introduced so-called caterpillar expressions (and automata) which can be used to model contexts in XML grammars. Since a caterpillar automaton is nothing more than a finite automaton whose accepted input words are interpreted as commands of the caterpillar (which then walks along the assumed syntax tree induced by the XML grammar), GI techniques may assist the XML designer also in designing caterpillar expressions describing contexts.

Ahonen [1,2] mentioned another possible application of GI for DTD generation, namely, assembly of (parts of) tagged documents from different sources (with different original DTDs). Hence, the assembled document is a transformation of one or more existing documents. The problem is to infer a common DTD. This assembly problem has also been addressed for XML recently [6] without referring to GI. The integration of both approaches seems to be promising.

Approximation. One possible objection against our approach could be to note that not *every* possible XML language can be inferred, irrespectively of the chosen distinguishing function, due to Lemma 5. We have observed [17] that, for any distinguishing function f and for every finite subset I_+ of an arbitrary regular set $R \subseteq \Sigma^*$, the language $\pi_f^{-1}\mathcal{A}_S(I_+)$ proposed by our algorithm for identifying f-distinguishable languages is the smallest language in (f, Σ)-DL which contains R. This sort of approximation property was investigated before by Kobayashi and Yokomori [26]. Due to the one-to-one correspondence between regular languages and XML languages induced by the notion of surface, this means that our proposed method for inferring XML languages can be used to approximate any given "spelled" XML language arbitrarily well.

The idea of incorporating GI techniques helping WWW applications also appears in [19,21].

References

1. H. Ahonen. Automatic generation of SGML content models. In *Electronic Publishing '96 (Palo Alto, California, USA)*, September 1996.

2. H. Ahonen. *Generating grammars for structured documents using grammatical inference methods*. Phd thesis. Also: Report A-1996-4, Department of Computer Science, University of Helsinki, Finland, 1996.
3. H. Ahonen, H. Mannila, and E. Nikunen. Forming grammars for structured documents: an application of grammatical inference. In R. C. Carrasco and J. Oncina, editors, *Proceedings of the Second International Colloquium on Grammatical Inference (ICGI-94): Grammatical Inference and Applications*, volume 862 of *LNCS/LNAI*, pages 153–167. Springer, 1994.
4. O. Altamura, F. Esposito, F. A. Lisi, and D. Malerba. Symbolic learning techniques in paper document processing. In P. Perner and M. Petrou, editors, *Machine learning and data mining in pattern recognition*, volume 1715 of *LNCS/LNAI*, pages 159–173. Springer, 1999.
5. D. Angluin. Inference of reversible languages. *Journal of the Association for Computing Machinery*, 29(3):741–765, 1982.
6. R. Behrens. A grammar based model for XML schema integration. In B. Lings and K. Jeffery, editors, *Advances in Databases, 17th British National Conference on Databases (BNCOD 17)*, volume 1832 of *LNCS*, pages 172–190. Springer, 2000.
7. R. Behrens and G. Buntrock. XML, eine Verwandte der Dyck-Sprachen. In *9. Theorietag der GI-Fachgruppe 0.1.5 Automaten und Formale Sprachen*, volume Preprint 12/99 of *Mathematische Schriften Kassel*, September 1999.
8. J. Berstel and L. Boasson. XML grammars. In N. Nielsen and B. Rovan, editors, *Mathematical Foundations of Computer Science (MFCS'2000)*, volume 1893 of *LNCS*, pages 182–191. Springer, 2000. Long Version as Technical Report IGM 2000-06, see `www-igm.univ-mlv.fr/~berstel/Recherche.html`.
9. H. Boström. Theory-guided induction of logic programs by inference of regular languages. In *Proc. of the 13th International Conference on Machine Learning*, pages 46–53. Morgan Kaufmann, 1996.
10. A. Brüggemann-Klein, S. Herrmann, and D. Wood. Context and caterpillars and structured documents. In E. V. Munson, C. Nicholas, and D. Wood, editors, *Principles of Digital Document Processing; 4th International Workshop (PODDP'98)*, volume 1481 of *LNCS*, pages 1–9. Springer, 1998.
11. A. Brüggemann-Klein and D. Wood. Caterpillars, context, tree automata and tree pattern matching. In G. Rozenberg and W. Thomas, editors, *Developments in Language Theory; Foundations, Applications, and Perspectives (DLT'99)*, pages 270–285. World Scientific, 2000.
12. CZ-Redaktion. Maschinenmenschen plauern per XML mit der Unternehmens-IT. *Computer Zeitung*, (50):30, December 2000.
13. P. Dupont and L. Miclet. Inférence grammaticale régulière: fondements théoriques et principaux algorithmes. Technical Report RR-3449, INRIA, 1998.
14. P. Fankhauser and Y. Xu. Markitup! An incremental approach to document structure recognition. *Electronic Publishing – Origination, Dissemination and Design*, 6(4):447–456, 1994.
15. H. Fernau. Identification of function distinguishable languages. In H. Arimura, S. Jain, and A. Sharma, editors, *Proceedings of the 11th International Conference Algorithmic Learning Theory ALT 2000*, volume 1968 of *LNCS/LNAI*, pages 116–130. Springer, 2000.
16. H. Fernau. k-gram extensions of terminal distinguishable languages. In *International Conference on Pattern Recognition (ICPR 2000)*, volume 2, pages 125–128. IEEE/IAPR, IEEE Press, 2000.

17. H. Fernau. Approximative learning of regular languages. Technical Report WSI–2001–2, Universität Tübingen (Germany), Wilhelm-Schickard-Institut für Informatik, 2001.
18. H. Fernau and J. M. Sempere. Permutations and control sets for learning non-regular language families. In A.L. Oliveira, editor, *Grammatical Inference: Algorithms and Applications, 5th International Colloquium (ICGI 2000)*, volume 1891 of *LNCS/LNAI*, pages 75–88. Springer, 2000.
19. D. Freitag. Using grammatical inference to improve precision in information extraction. In *Workshop on Grammatical Inference, Automata Induction, and Language Acquisition (ICML'97), Nashville, TN*, 1997. Available through: `http://www.univ-st-etienne.fr/eurise/pdupont/mlworkshop.html#proc`.
20. P. García and E. Vidal. Inference of k-testable languages in the strict sense and applications to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12:920–925, 1990.
21. T. Goan, N. Benson, and O. Etzioni. A grammatical inference algorithm for the World Wide Web. In *Working Notes of the AAAI-96 Spring Symposium on Machine Learning in Information Access*, 1996.
22. E. M. Gold. Language identification in the limit. *Information and Control (now Information and Computation)*, 10:447–474, 1967.
23. J. Gregor. Data-driven inductive inference of finite-state automata. *International Journal of Pattern Recognition and Artificial Intelligence*, 8(1):305–322, 1994.
24. C. de la Higuera. Current trends in grammatical inference. In F. J. Ferri et al., editors, *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR+SPR'2000*, volume 1876 of *LNCS*, pages 28–31. Springer, 2000.
25. T. Hu and R. Ingold. A mixed approach toward an efficient logical structure recognition. *Electronic Publishing – Origination, Dissemination and Design*, 6(4):457–468, 1994.
26. S. Kobayashi and T. Yokomori. Learning approximately regular languages with reversible languages. *Theoretical Computer Science*, 174(1–2):251–257, 1997.
27. E. Mäkinen. Inferring regular languages by merging nonterminals. *International Journal of Computer Mathematics*, 70:601–616, 1999.
28. T. Mitchell. Machine Learning. McGraw-Hill, 1997.
29. T. Mitchell. Machine learning and data mining. *Communications of the ACM*, 42:31–36, 1999.
30. S. Muggleton and L. De Raedt. Inductive logic programming: theory and methods. *Journal of Logic Programming*, 20:629–679, 1994.
31. V. Radhakrishnan. *Grammatical Inference from Positive Data: An Effective Integrated Approach.* PhD thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay (India), 1987.
32. V. Radhakrishnan and G. Nagaraja. Inference of regular grammars via skeletons. *IEEE Transactions on Systems, Man and Cybernetics*, 17(6):982–992, 1987.
33. G. Semeraro, F. Esposito, and D. Malerba. Learning contextual rules for document understanding. In *Proceedings of the 10th IEEE Conference on Artificial Intelligence for Applications*, pages 108–115, 1994.
34. R. E. Tarjan. Efficiency of a good but not linear set union algorithm. *Journal of the Association for Computing Machinery*, 22(2):215–225, 1975.
35. P. T. Wood. Rewriting XQL queries on XML repositories. In B. Lings and K. Jeffery, editors, *Advances in Databases, 17th British National Conference on Databases (BNCOD 17)*, volume 1832 of *LNCS*, pages 209–226. Springer, 2000.

First-Order Rule Induction for the Recognition of Morphological Patterns in Topographic Maps

D. Malerba, F. Esposito, A. Lanza, F.A. Lisi

Dipartimento di Informatica, University of Bari
via Orabona 4, 70126 Bari, Italy
{malerba | esposito | lanza | lisi}@di.uniba.it

Abstract. Information given in topographic map legends or in GIS models is often insufficient to recognize interesting geographical patterns. Some prototypes of GIS have already been extended with a knowledge-base and some reasoning capabilities to support sophisticated map interpretation processes. Nevertheless, the acquisition of the necessary knowledge is still an open problem to which machine learning techniques can provide a solution. This paper presents an application of first-order rule induction to pattern recognition in topographic maps. Research issues related to the extraction of first-order logic descriptions from vectorized topographic maps are introduced. The recognition of morphological patterns in topographic maps of the Apulia region is presented as a case study.

1 Introduction

Handling digitized maps raises several research issues for the field of pattern recognition. For instance, raster-to-vector conversion of maps has received increasing attention in the community of graphics recognition [6]. In fact, obtaining vector data from a paper map is a very expensive and slow process, which often requires manual intervention. While supporting the map acquisition process is important, it is equally useful and even more challenging to automate the *interpretation* of a map in order to locate some geographic objects and their relations [12]. Indeed information given by map legends or given as basis of data models in Geographical Information Systems (GIS) is often insufficient to recognize not only *geographical objects* relevant for a certain application, but also *patterns* of geographical objects which geographers, geologists and town planners are interested in. Map interpretation tasks such as the detection of morphologies characterizing the landscape, the selection of important environmental elements, both natural and artificial, and the recognition of forms of the territorial organization require abstraction processes and deep domain knowledge that only human experts have.

Several studies show the difficulty of map interpretation tasks. For instance, a study on the drawing instructions of Bavarian cadastral maps (scale 1:5000) pointed out that symbols for road, pavement, roadside, garden and so on were defined neither in the legend nor in the GIS-model of the map [16]. In a previous work in cooperation with researchers from the Town Planning Department of the Polytechnic of Bari, an

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 88–101, 2001.

environmental planning expert system was developed for administrators responsible for urban planning [2], [1]. The system was able to provide them with appropriate suggestions but presumed that they had good skills in reading topographic maps to detect some important ground morphology elements, such as system of cliffs, ravines, and so on. These are some examples of morphological patterns that are very important in many civil and military applications but never explicitly represented in topographic maps or in a GIS-model.

Empowering GIS with advanced pattern recognition capabilities would support effectively map readers in map interpretation tasks. Some prototypes of GIS have already been extended with a knowledge-base and some reasoning capabilities in order to support sophisticated map interpretation processes [20]. Nevertheless, these systems have a limited range of applicability for a variety of reasons mainly related to the knowledge acquisition bottleneck.

A solution to these difficulties can come from machine learning. In this paper we present an application of first-order rule induction to pattern recognition in topographic maps. Reseach issues related to the extraction of first-order logic descriptions from vectorized topographic maps are introduced. The task of topographic map interpretation as a whole is supported by INGENS (Inductive Geographic Information System), a prototypical GIS extended with a training facility and an inductive learning capability [16]. In INGENS, each time a user wants to retrieve geographic complex objects or patterns not explicitly modeled in the Map Repository, he/she can prospectively train the system to the recognition task within a special user view. Training is based on a set of examples and counterexamples of geographic concepts of interest to the user (e.g., ravine or steep slopes). Such (counter-) examples are provided by the user who detects them on stored maps by applying browsing, querying and displaying functions of the GIS interface. The symbolic representation of the training examples is automatically extracted from maps by the module Map Descriptor. The module Learning Server implements one or more inductive learning algorithms that can generate models of geographic objects from the chosen representations of training examples. In this paper, we will focus our presentation on the first-order rule induction algorithm ATRE [15].

The data model for the Map Repository of INGENS is described in the next section. In Section 3, the feature extraction algorithms implemented in the Map Descriptor are sketched. Section 4 is devoted to the first-order rule induction algorithm ATRE made available in the Learning Server. A case study, namely the recognition of relevant morphological patterns on topographic maps of the Apulia region, is presented and discussed in Section 5. Conclusions and future work are reported in Section 6.

2 A Data Model for Topographic Maps

Many GIS store topographic maps. In the Map Repository of INGENS each map is stored according to a *hybrid tessellation – topological model*. The tessellation model follows the usual topographic practice of superimposing a regular grid on a map in order to simplify the localization process. Indeed each map in the repository is

divided into square cells of the same size. For each cell the raster image in GIF format is stored together with its coordinates and component objects. In the topological model of each cell it is possible to distinguish two different structural hierarchies: *physical* and *logical*.

The physical hierarchy describes the geographical objects by means of the most appropriate physical entity, that is: point, line or region. In different maps of the same geographical area, the same object may have different physical representations. For instance, a road can be represented as a line on a small-scale map, or as a region on a large-scale map. Points are described by their spatial coordinates, while (broken) lines are characterized by the list of line vertices, and regions are represented by their boundary line. Some topological relationships between points, lines and regions are modeled in the conceptual design, namely points inside a region or on its border, and regions disjoining/meeting/overlapping/containing/equaling/covering other regions. The meaning of the topological relationships between regions is a variant of that reported in the 9-intersection model by Egenhofer and Herring [7], in order to take into account problems due to approximation errors.

The logical hierarchy expresses the semantics of geographical objects, independent of their physical representation. Since the conceptual data model has been designed to store topographic maps, the logical entities concern geographic layers such as hydrography, orography, land administration, vegetation, administrative (or political) boundary, ground transportation network, construction and built-up area. Each of them is, in turn, a generalization meaning that, for instance, an administrative boundary must be classified in one of the following classes: city, province, county or state.

3 Feature Extraction from Vectorized Topographic Maps

In INGENS the content of a map cell is described by means of a set of *features*. Here the term feature is intended as a characteristic (property or relationship) of a geographical entity. This meaning is similar to that commonly used in Pattern Recognition (PR) and differs from that attributed by people working in the field of GIS, where the term feature denotes the unit of data by which a geographical entity is represented in computer systems and, according to the OGC terminology, is modelled through a series of properties [17], [21].

In PR, feature is a synonym for discriminatory property of objects which have to be recognised and classified. Obviously, the number of features needed to successfully perform a given recognition task depends on the discriminatory qualities of the chosen features. However, the problem of *feature selection* (i.e. what discriminatory features to select), is usually complicated by the fact that the most important features are not necessarily easily measurable. *Feature extraction* is an essential phase which follows the segmentation in the classical recognition methodology [11]. In PR, features are classified into three categories according to their nature: *physical*, *structural*, and *mathematical* [22]. The first two categories are used primarily in the area of image processing, while the third one includes statistical

means, correlation coefficients and so on. In map interpretation tasks a different category of features is required, namely *spatial* features.

Tables 1 and 2 show a taxonomy of spatial features that can be to be extracted from vectorized maps. The first distinction to be made concerns the *type* of feature: it can be an *attribute*, that is a property possessed by the spatial object, or a *relation* that holds among the object itself and other objects. Spatial relationships among geographic objects are actually conditions on object positions.

According to the *nature* of the feature, it is possible to distinguish among:

- *Locational* features, when they concern the position of the objects. The position of a geographic object will be represented by numeric values expressing coordinates for example in latitude/longitude or in polar coordinates or others.
- *Geometric* features, when they depend on some computation of metric/distance. Area, perimeter, length are some examples. Their domain is typically numeric.
- *Topological* features (actually only a relation can be topological), when they are preserved under topological transformations, such as translation, rotation, and scaling. Topological features are generally represented by nominal values.
- *Directional* features, when they concern orientation (e.g., north, north-east, and so on). Generally, a directional feature is represented by means of nominal values.

Clearly, a geo-referenced object also has aspatial features, such as the name, the layer label, and the temperature. Many other features can be extracted from maps, some of which are hybrid in the sense that merge properties of two or more categories. For instance, the features that express the conditions of parallelism and perpendicularity of two lines are both topological and geometrical. They are topological since they are invariant with respect to translation, rotation and stretching, while they are geometrical since their semantics is based on the size of their angle of incidence. Another example of hybrid spatial feature is represented by the relation of "faraway-west", whose semantics mixes both directional and geometric concepts. Finally, some features might mix spatial relations with aspatial properties, such as the feature that describes coplanar roads by combining the condition of parallelism with information on the type of spatial objects.

The problem of extracting features from maps has been mainly investigated in the fields of document processing and graphics recognition, nevertheless most of the

Table 1. A classification of attributive features.

ATTRIBUTES			
SPATIAL			**ASPATIAL**
LOCATIONAL	**GEOMETRIC**	**DIRECTIONAL**	Name Layer Type Others (temperature, no. inhabitants, …)
Co-ordinate (x,y) of a point (centroid, extremal points, bounding rectangles, …)	▪ Area ▪ Perimeter ▪ Length of axes Other shape properties	Orientation of major axis	

Table 2. A classification of relational features.

RELATIONS			
SPATIAL			**ASPATIAL**
GEOMETRIC	**TOPOLOGICAL**	**DIRECTIONAL**	▪ Instance-of ▪ Hierarchical relation (sub-type, super-type) ▪ Aggregation/ Composition
▪ Distance ▪ Angle of incidence	▪ Region-to-Region ▪ Region-to-Line ▪ Region-to-Point ▪ Line-to-Line ▪ Line-to-Point ▪ Point-to-Point	▪ Neighbouring relations	

work reported in the literature concerns raster maps, where the issues are how to isolate text from graphics as in the work by Pierrot *et al.* [18], or how to extract particular geographical objects, such as contour lines as in [6], or points and lines as in the work by Yamada *et al.* [23] or land-use classes using thematic maps as in [3].

The lack of works on vectorized representations can be attributed to the main usage of topographic maps made in the field of GIS: only for rendering purposes. The rare applications to vectorized maps reported in the literature refer to cadastral maps, as in [5].

A first application of feature extraction algorithms to vectorized topographic maps can be found in the work by Esposito *et al.* [8]. This work is a natural evolution of the collaboration already established between a research group on Machine Learning of the University of Bari with the Town Planning Department of the Polytechnic of Bari in order to develop an expert system for environmental planning [2], [1]. For environmental planning tasks, fifteen features were specified with the help of domain experts (see Table 3). Being quite general, they can also be used to describe maps on different scales. In INGENS they are extracted by the module *Map Descriptor*, which generates first-order logic descriptions of the maps stored in the Map Repository.

Actually, feature extraction procedures working on vectorized maps are far from being a simple "adaptation" of existing graphics recognition algorithms. In fact, the different data representation (raster vs. vector) makes the available algorithms totally unsuitable to vectorized maps, as it is the case of all filters based on the mathematical morphology [23]. Each feature to be extracted needs a specific procedure to be developed basing upon the geometrical, topological and topographical principles, which are involved in the semantics of that feature.

For instance, the relation *distance* between two "parallel" lines is computed by means of the following algorithm. Let O_1 and O_2 be two geographical linear objects represented by n and m coordinate pairs, respectively. Without loss of generality, let us assume that $n \geq m$. The algorithm first computes $d\min_h$ as the minimum distance

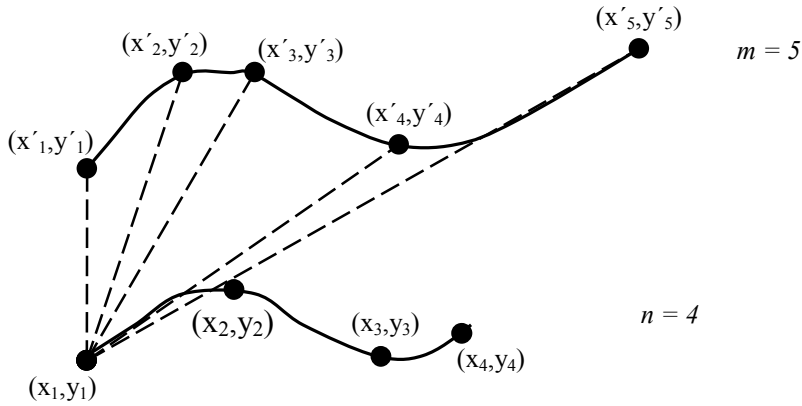


Fig. 1. Computation of the distance between two "parallel" lines.

between the h-th point of O_1 and any point of O_2 (see Figure 1). Then, the distance between O_1 and O_2 is computed as follows:

$$distance = \frac{\sum_{h=1}^{n} d\min_h}{n} \quad (1)$$

The complexity of this simple feature extraction algorithm is O(mn) though less computationally expensive solutions can be found by applying multidimensional access methods [10].

The descriptions obtained for each cell are quite complex, since some cells contain dozens of geographic objects of various types. For instance, the cell shown in Figure 2 contains one hundred and eighteen distinct objects, and its complete description is a clause with more than one thousand literals in the body.

4 The Induction of First-Order Rules with ATRE

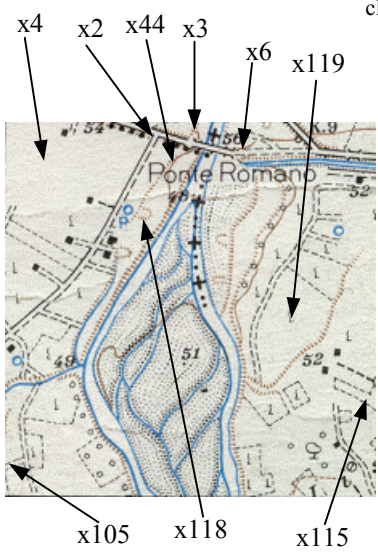
Sophisticated end users may train INGENS to recognize geographical patterns that are not explicitly modeled in the Map Repository. To support this category of users, the module *Learning Server* places some inductive learning systems at their disposal. We will focus our attention on the first-order rule induction algorithm ATRE [14].

The distinguishing feature of ATRE is that it can induce recursive logical theories from a set of training examples. Here the term *logical theory* (or simply *theory*) denotes a set of first-order definite clauses. An example of logical theory is the following:

downtown(X) ? high_business_activity(X), onthesea(X).
residential(X) ? close_to(X,Y), downtown(Y), low_business_activity(X).
residential(X) ? close_to(X,Y), residential(Y), low_business_activity(X).

Table 3. Features extracted for the generation of map descriptions.

Feature	Meaning	Type	Domain	
			Type	Values
CONTAIN(X,Y)	Cell X contains object Y	Topologic relation	boolean	{true, false}
TYPE_OF(Y)	Object Y type	Aspatial attribute	nominal	33 nominal values
SUBTYPE_OF(Y)	Specialization of object Y type	Aspatial attribute	nominal	101 nominal values that are specializations of type_of domain
COLOR(Y)	Object Y color	Aspatial attribute	nominal	{blue, brown, black}
AREA(Y)	Object Y area	Geometrical attribute	linear	[0..MAX_AREA]
DENSITY(Y)	Object Y density	Geometrical attribute	ordinal	Symbolic names chosen by expert user
EXTENSION(Y)	Object Y extension	Geometrical attribute	linear	[0..MAX_EXTENTION]
GEOGRAPHIC_DIRECTION(Y)	Geographic direction of Y	Directional attribute	nominal	{north, east, north_west, north_east}
LINE_SHAPE(Y)	Shape of the linear object Y	Geometrical attribute	nominal	{straight, curvilinear, cuspidal}
ALTITUDE(Y)	Altitude of Y	Geometrical attribute	linear	[0.. MAX_ALTITUDE]
LINE_TO_LINE(Y,Z)	Spatial relation between two lines Y and Z	Hybrid relation	nominal	{almost parallel, almost perpendicular}
DISTANCE(Y,Z)	Distance between objects Y and Z	Geometrical relation	linear	[0..MAX_DISTANCE]
REGION_TO_REGION(Y,Z)	Spatial relation between two regions Y and Z	Topological relation	nominal	{disjoint, meet, overlap, covers, covered_by, contains, equal, inside}
LINE_TO_REGION(Y,Z)	Spatial relation between a line Y and a region Z	Hybrid relation	nominal	{along_edge, intersect}
POINT_TO_REGION(Y,Z)	Spatial relation between a point Y and a region Z	Topological relation	nominal	{inside, outside, on_boundary, on_vertex}



class(x1)=fluvial_landscape ?
contain(x1,x2)=true, contain(x1,x3)=true, …,
contain(x1,x119)=true, type_of(x2)=main_road,
type_of(x3)=slope, …, type_of(x119)=vegetation,
color(x2)=black, color(x3)=brown,…,
color(x119)=black, trend(x2)=straight,
trend(x3)=straight, …, trend(x118)=curvilinear,
extension(x2)=340.352, extension(x8)=134.959, …,
extension(x119)=94.162,
geographic_direction(x2)=north_west,
geographic_direction(x8)=north_east, …,
geographic_direction(x114)=north_east,
shape(x28)=non_cuspidal, shape(x44)=non_cuspidal,
…, shape(x92)=non_cuspidal, density(x4)=low,
density(x17)=low, …, density(x115)=low,
line_to_line(x2,x6)=almost_perpendicular,
line_to_line(x8,x6)=almost_perpendicular, …,
line_to_line(x5,x114)=almost_parallel,
distance(x6,x10)=311.065, distance(x6,x11)=466.723,
…, distance(x105,x114)=536.802

Fig. 2. A partial logical description of a cell. The constant *x1* represents the whole cell, while all other constants denote the one hundred and eighteen enclosed objects. Distances and extensions are expressed in meters.

It expresses sufficient conditions for the two concepts of "main business center of a city" and "residential zone," which are represented by the unary predicates *downtown* and *residential*, respectively.

The learning problem solved by ATRE can be formulated as follows:

Given

- a set of concepts C_1, C_2, ?, C_r to be learned,
- a set of observations O described in a language L_O,
- a background knowledge BK described in a language L_{BK},
- a language of hypotheses L_H,
- a generalization model Γ over the space of hypotheses,
- a user's preference criterion PC,

Find

a (possibly recursive) logical theory T for the concepts C_1, C_2, ?, C_r, such that T is complete and consistent with respect to O and satisfies the preference criterion PC.

The *completeness* property holds when the theory T explains all observations in O of the r concepts C_i, while the *consistency* property holds when the theory T explains no counter-example in O of any concept C_i. The satisfaction of these properties guarantees the correctness of the induced theory with respect to O.

As regards the representation languages L_O, L_{BK}, L_H, the basic component is the *literal*, which takes two distinct forms:

$f(t_1, ?, t_n) = Value$ (*simple literal*) $f(t_1, ?, t_n)$ $?[a..b]$ (*set literal*),

where f and g are function symbols called *descriptors*, t_i's and s_i's are terms, and $[a..b]$ is a closed interval. Descriptors can be either *nominal* or *linear*, according to the ordering relation defined on its domain values. Some examples of literals are:

color(X)=blue, *distance(X,Y)=63.9*, *width(X)?[82.2 .. 83.1]*, and c*lose_to(X,Y)=true*. The last example points out the lack of predicate symbols in the representation languages adopted by ATRE. Thus, the first-order literals *p(X,Y)* and ?*p(X,Y)* will be represented as $f_p(X,Y)=true$ and $f_p(X,Y)=false$, respectively, where f_p is the function symbol associated to the predicate *p*. Henceforth, for the sake of simplicity, we will adopt the usual notation *p(X,Y)* and ?*p(X,Y)*. Furthermore, the interval *[a..b]* in a set literal $f(X_1, ..., X_n)?[a..b]$ is computed according to the same information theoretic criterion used in INDUBI/CSL [13].

Observations in ATRE are represented as ground multiple-head clauses, called *objects*, which have a conjunction of simple literals in the head. Multiple-head clauses present two main advantages with respect to definite clauses: higher comprehensibility and efficiency. The former is basically due to the fact that multiple-head clauses provide us with a compact description of multiple properties to be predicted in a complex object such as those we may have in map interpretation. The second advantage derives from the possibility of having a unique representation of known properties shared by a subset of observations.

The *background knowledge* defines any relevant problem domain knowledge. It is expressed by means of *linked*, *range-restricted* definite clauses [4] with simple and set literals in the body and one simple literal in the head. The same constraints are applied to the language of hypotheses.

ATRE implements a novel approach to the induction of recursive theories [9]. To illustrate how the main procedure works, let us consider the following instance of the learning problem:

Observations O$_1$ $downtown(zone_1)?\ ?residential(zone_1)?\ residential(zone_2)\ ?$
$?downtown(zone_2)\ ?\ ?downtown(zone_3)\ ?\ residential(zone_4)\ ?$
$?downtown(zone_4)\ ?\ ?downtown(zone_5)\ ?\ ?residential(zone_5)\ ?$
$?residential(zone_6)\ ?\ downtown(zone_7)\ ?\ ?residential(zone_7) \leftarrow$
$onthesea(zone_1), high\_business\_activity(zone_1),$
$close\_to(zone_1,zone_2),$
$low\_business\_activity(zone_2), close\_to(zone_2,zone_4),$
$adjacent(zone_1,zone_3), onthesea(zone_3),$
$low\_business\_activity(zone_3), low\_business\_activity(zone_4),$
$close\_to(zone_4,zone_5), high\_business\_activity(zone_5),$
$adjacent(zone_5,zone_6), low\_business\_activity(zone_6),$
$close\_to(zone_6,zone_8), low\_business\_activity(zone_8),$
$close\_to(zone_1,zone_7), onthesea(zone_7),$

BK $close\_to(X,Y) \leftarrow adjacent(X,Y)$
$close\_to(X,Y) \leftarrow close\_to(Y,X)$

Concepts C$_1$ $downtown(X)=true$
$residential\_zone(X)=true$

C$_2$

PC Minimize/maximize negative/positive examples explained by the theory

The first step towards the generation of inductive hypotheses is the saturation of all observations with respect to the given *BK* [19]. In this way, information that was

implicit in the observation, given the background knowledge, is made explicit. In the example above, the saturation of O_1 involves the addition of the nine literals logically entailed by BK, that is *close_to(zone$_2$, zone$_1$)*, *close_to(zone$_1$, zone$_3$)*, *close_to(zone$_3$, zone$_1$)*, *close_to(zone$_7$, zone$_1$)*, *close_to(zone$_4$, zone$_2$)*, *close_to(zone$_5$, zone$_4$)*, *close_to(zone$_5$, zone$_6$)*, *close_to(zone$_6$, zone$_5$)*, and *close_to(zone$_8$, zone$_6$)*.

Initially, all positive and negative examples are generated for every concept to be learned, the learned theory is empty, while the set of concepts to be learned contains all C_i. With reference to the above input data, the system generates two positive examples for C_1 (*downtown(zone$_1$)* and *downtown(zone$_7$)*), two positive examples for C_2 (*residential(zone$_2$)* and *residential(zone$_4$)*), and eight negative examples equally distributed between C_1 and C_2 (*?downtown(zone$_2$)*, *?downtown(zone$_3$)*, *?downtown(zone$_4$)* ? *?downtown(zone$_5$)*, *?residential(zone$_1$)*, *?residential(zone$_5$)*, *?residential(zone$_6$)*, *?residential(zone$_7$)*).

Once the observations have been saturated and examples have been generated, the separate-conquer loop starts. The step of *parallel conquer* generates a set of consistent clauses, whose minimum number is defined by the user. Since clauses are consistent, they should explain no negative example. For instance, by requiring the generation of at least one consistent clause with respect to the examples above, this procedure returns the following set of clauses:

downtown(X) ? onthesea(X), high_business_activity(X).
downtown(X) ? onthesea(X), adjacent(X,Y).
downtown(X) ? adjacent(X,Y), onthesea(Y).

The first of them is selected according to the preference criterion (procedure *find_best_clause*). Actually, the hypothesis space of the concept *residential* has been simultaneously explored, but at the time in which the three consistent clauses for the concept *downtown* have been found, no consistent clause for *residential* has been discovered yet. Thus the parallel conquer step stops since the number of consistent clauses is greater than one.

Since the addition of a consistent clause to the partially learned theory may lead to an augmented, inconsistent theory, it is necessary to verify the *global consistence* of the learned theory and eventually reformulate the theory in order to recover the consistency property without repeating the learning process from scratch. The learned clause is used to saturate again the observation. Continuing the previous example, the two literals added to O_1 are *downtown(zone$_1$)* and *downtown(zone$_7$)*. This operation enables ATRE to generate also definitions of the concept *residential* that depend on the concept *downtown*. Indeed, at the second iteration of the separate-conquer cycle, the parallel conquer step returns the clause:

residential(X) ? close_to(X,Y), downtown(Y), low_business_activity(X).

and by saturating again the observation with both learned clauses, it becomes possible to generate a recursive clause at the third iteration, namely

residential(X) ? close_to(X,Y), residential(Y), low_business_activity(X).

The separate step consists of tagging positive examples explained by the current learned theory, so that they are no longer considered for the generation of new clauses. The separate-conquer loop terminates when all positive examples are tagged, meaning that the learned theory is complete as well as consistent.

5 The Recognition of Morphological Patterns in Topographic Maps: A Case Study

The first-order rule induction algorithm ATRE has been applied to the recognition of four morphological patterns in topographic maps of the Apulia region, Italy, namely *regular grid system of farms*, *fluvial landscape*, *system of cliffs* and *royal cattle track*. Such patterns are deemed relevant for the environmental protection, and are of interest to town planners. A regular grid system of farms is a particular model of rural space organization that originated from the process of rural transformation. The fluvial landscape is characterized by the presence of waterways, fluvial islands and embankments. The system of cliffs presents a number of terrace slopes with the emergence of blocks of limestone. A royal cattle track is a road for transhumance that can be found exclusively in the South-Eastern part of Italy.

The territory considered in this application covers 131 km^2 in the surroundings of the Ofanto River, spanning from the zone of Canosa to the Ofanto mouth. More precisely, the examined area is covered by five map sheets on a scale of 1:25000 produced by the IGMI (Ofanto mouth – 165 II S.W., Barletta 176 I N.W., Canne della Battaglia – 176 IV N.E., Montegrosso 176 IV S.E., Canosa 176 IV S.W.).

The maps have been segmented into square observation units of 1 Km^2 each. The choice of the gridding step, which is crucial for the recognition task, has been made using the advice of a team of fifteen geomorphologists and experts in environmental planning, giving rise to a one-to-one mapping between observation units of the map and cells in the database.

Thus, the problem of recognizing the four morphological patterns can be reformulated as the problem of labeling each cell with at most one of four labels. Unlabelled cells are considered uninteresting for environmental protection.

As previously mentioned, ATRE extends the system INGENS with a training functionality and an inductive learning capability in order to overcome the difficulties related to the acquisition of operational definitions for the recognition task. ATRE was trained according to the experimental design briefly presented below. One hundred and thirty-one cells were selected, each of which was described in the symbolic language illustrated in the previous Section and assigned to one of the following five classes: system of farms, fluvial landscape, system of cliffs, royal cattle track and other. The last class simply represents "the rest of the world," and no classification rule is generated for it. Indeed, its assigned cells are not interesting for the problem of environmental protection being studied, and they are always used as negative examples when ATRE learns classification rules for the remaining classes. Forty-five cells from the map of Canosa were selected to train the system, while the remaining eighty-six cells were randomly selected from the four maps of the Ofanto mouth, Barletta, Canne della Battaglia and Montegrosso. Training observations represent about 35% of the total experimental data set. An example of partial logical description of a training cell is shown in Figure 2.

A fragment of the logical theory induced by ATRE is reported below:

class(X1) = fluvial_landscape ?contain(X1,X2), color(X2)=blue,
type_of(X2)=river,trend(X2)=curvilinear, extension(X2)?[325.00..818.00].
class(X1) = fluvial_landscape ?contain(X1,X2), type_of(X2)=river, color(X2)=blue,

relation(X3,X2)=almost_perpendicular,
extension(X2)?[615.16..712.37],trend(X3)=straight.
class(X1)=system_of_farms ?contain(X1,X2), color(X2)=black,
relation(X2,X3)=almost_perpendicular,
relation(X3,X4)=almost_parallel,type_of(X4)=interfarm_road,
geographic_direction(X4)=north_est,
extension(X2)?[362.34 .. 712.25], color(X3)=black, type_of(X3)=farm_road,
color(X4)=black.

The first two clauses explain all training observations of fluvial landscape. In particular, the first states that cells labeled as *fluvial_landscape* contain a long, curvilinear, blue object of type river, while the second clause states that cells concerning a fluvial landscape may also present a long, straight, blue object that is perpendicular to another object (presumably, a bridge). The third clause refers to the system of farms. From the training observations, the machine learning system induced the following definition: "There are two black objects, namely an interfarm road (*X4*) and a farm road (*X3*), which run almost parallel to the north-east, and are both perpendicular to a long black object". This definition of system of farms is not complete since it includes other clauses that ATRE actually generated but are not reported in this paper. It is easy to see that the classification rules are intelligible and meaningful. Some experimental results obtained in a previous work are reported in [8].

By matching these rules with logical descriptions of other map cells it is possible to automate the recognition of complex geographical objects or geographical patterns that have not been explicitly modeled by a set of symbols.

6 Conclusions

Automated map interpretation is a challenging application domain for pattern recognition. Knowledge of the meaning of symbols reported in the map legends is not generally sufficient to recognize interesting geographical complex objects or patterns on a map. Moreover, it is quite difficult to describe such patterns in a machine-readable format. That would be tantamount to providing GIS with an operational definition of abstract concepts often reported in texts and specialist handbooks. In order to enable the automation of map interpretation tasks in GIS, a new approach has been proposed in this paper. The idea is to ask GIS users for a set of classified instances of the patterns that interest them, and then apply a first-order rule induction algorithm to generate the operational definitions for such patterns. These definitions can be either used to recognize new occurrences of the patterns at hand in the Map Repository. An application to the problem of Apulian map interpretation has been reported in this paper in order to illustrate the advantages of the proposed approach.

This work is still in progress and many problems have to be solved. As for the data model for topographic maps, the segmentation of a map in a grid of suitably sized cells is a critical factor, since over-segmentation leads to a loss of recognition of global effects, while under-segmentation leads to large cells with an unmanageable number of components. To cope with the first problem, it is necessary to consider the *context* of a cell, that is the neighboring cells, both in the training phase and in the

recognition phase. To solve problems caused by under-segmentation it is crucial to provide users with appropriate tools that hide irrelevant information in the cell description. Indeed, a set of generalization and abstraction operators will be implemented in order to simplify the complex descriptions currently produced by the *Map Descriptor*.

As for the algorithm ATRE, we plan to further investigate the influence of both the representation and the content of observations in the training set on experimental results. Case studies stressing the capability of autonomously discovering concept dependencies should also be faced.

Acknowledgements

This work is in partial fulfillment of the research objectives set by the IST project SPIN! (Spatial Mining for Data of Public Interest) funded by the European Union (`http://www.ccg.leeds.ac.uk/spin/`).

References

1. Barbanente, A., Borri, D., Esposito, F., Leo, P., Maciocco, G., Selicato, F.: Automatically acquiring knowledge by digital maps in artificial intelligence planning techniques. In: Frank, A.U., Campari, I., Formentini, U. (eds.): Theories and Methods of Spatio-Temporal Reasoning. Lecture Notes in Artificial Intelligence, Vol. 482. Springer-verlag, Berlin (1992) 89-100.
2. Borri D. et al.: Green: Building an Operational Prototype of Expert System for Planning Control in Urban Environment. Proceedings of the European Conference of the Regional Science Association, Instanbul (1990).
3. Centeno J.S.: Segmentation of Thematic Maps Using Colour and Spatial Attributes. In: Tombre, K., Chhabra, A.K. (eds.): Graphics Recognition Algorithms and Systems. Lecture Notes in Computer Science, Vol. 1389, Springer-Verlag, Berlin (1998) 221-230.
4. de Raedt, L.: Interactive theory revision: An inductive logic programming approach. Academic Press, London (1992).
5. den Hartog, J. , Holtrop, B.T., de Gunst, M.E., Oosterbroek E.P.: Interpretation of Geographic Vector-Data in Practice. In Chhabra, A.K., Dori, D. (eds.): Graphics Recognition Recent Advances. Lecture Notes in Computer Science, Vol. 1941, Springer-Verlag, Berlin (1999) 50-57
6. Dupon, F., Deseilligny, M.P., Gondran, M.: Automatic Interpretation of Scanned maps: Reconstruction of Contour Lines. In: Tombre, K., Chhabra, A.K. (eds.): Graphics Recognition: Algorithms and Systems. Lecture Notes in Computer Science, Vol. 1389, Springer-Verlag, Berlin (1998) pp. 1-8.
7. Egenhofer, M. J., Herring, J.R.: Categorising topological spatial relations between point, line, and area objects. In: Egenhofer, M.J., Mark, D.M., Herring, J.R. (eds.): The 9-intersection: formalism and its use for natural language spatial predicates. Technical Report NCGIA 94-1, Santa Barbara, (1994).
8. Esposito, F., Lanza, A., Malerba, D., Semeraro, G.: Machine learning for map interpretation: an intelligent tool for environmental planning. Applied Artificial Intelligence 11(7-8) (1997) 673-695.

9. Esposito, F., Malerba, D., Lisi, F.A.: Induction of Recursive Theories in the Normal ILP Setting: Issues and Solutions. In: Cussens, J., Frish, A. (eds.): Inductive Logic Programming, Vol. 1866, Springer, Berlin (2000) 93-111.
10. Gaede, V., Günther O.: Multidimensional Access Methods, ACM Computing Surveys, 30(2) (1998) 170-231.
11. Haralick, R.M., Shapiro, L.G.: Computer and Robot Vision, Addison-Wesley, Reading, MA (1992).
12. Keates, J. S.: Map understanding. Second Editon. Longman, Edinburgh (1996).
13. Malerba, D., Esposito, F., Semeraro, G. , Caggese, S.: Handling Continuous Data in Top-down Induction of First-order Rules. In: Lenzerini, M. (ed.): AI*IA 97: Advances in Artificial Intelligence, Lecture Notes in Artificial Intelligence,Vol. 1321. Springer, Berlin (1997) 24-35.
14. Malerba, D., Esposito, F.,Lisi, F.A.: Learning Recursive Theories with ATRE. In: Prade, H. (ed.): Proceedings of the 13th European Conf. on Artificial Intelligence, Wiley, Chichester (1998) 435-439.
15. Malerba, D., Esposito, F., Lanza, A., Lisi, F.A.: Discovering Geographic Knowledge: The INGENS System. In: Ras, Z.W., Ohsuga, S. (eds.): Foundations of Intelligent Information Systems, Lecture Notes in Artificial Intelligence, 1321, Springer, Berlin (2000) 40-48.
16. Mayer, H.: Is the knowledge in map-legends and GIS-models suitable for image understanding? International Archives of Photogrammetry and Remote Sensing 30(4) (1994) 52-59.
17. Open GIS Consortium: The OpenGIS Abstract Specification (1996), http://www.opengis.org/public/abstract.html.
18. Pierrot Deseilligny, M., Le Men, H., Stamon, G.: Characters String Recognition on Maps: A Method for High Level Reconstruction. Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 1. (1995) 249-252.
19. Rouveirol, C.: Flattening and saturation: Two representation changes for generalization. Machine Learning 14(2) (1994) 219-232.
20. Smith, T., Donna, P., Sudhakar, M., Pankaj, A.: KBGIS-II: A Knowledge-Based Geographic Information System. International Journal of Geographic Information Systems 1(2) (1987) 149-172.
21. Sondheim, M., Gardels, K., Buehler, K.: GIS Interoperability. In: Longley, P.A., Goodchild, M.F., Maguire, D.J., Rhinds, D.W. (eds.): Geographical Information Systems, Principles and Technical Issues, Vol. 1. John Wiley & Sons (1999) 347-358.
22. Tou, J. T., Gonzales, R. C.: Pattern Recognition Principles. Addison-Wesley, Reading, MA (1974).
23. Yamada, H. , Yamamoto, K., Hosokawa, K.: Directional Mathematical Morphology and Reformalized Hough Transformation for the Analysis of Topographic Maps, IEEE Transactions on Pattern Analysis and Machine Intelligence, 15(4) (1993) 380-387.

Concepts Learning with Fuzzy Clustering and Relevance Feedback

Bir Bhanu and Anlei Dong

Center for Research in Intelligent Systems
University of California, Riverside, California 92521, USA
{bhanu, adong}@cris.ucr.edu

Abstract. In recent years feedback approaches have been used in relating low-level image features with concepts to overcome the subjective nature of the human image interpretation. Generally, in these systems when the user starts with a new query, the entire prior experience of the system is lost. In this paper, we address the problem of incorporating prior experience of the retrieval system to improve the performance on future queries. We propose a semi-supervised fuzzy clustering method to learn class distribution (meta knowledge) in the sense of high-level concepts from retrieval experience. Using fuzzy rules, we incorporate the meta knowledge into a probabilistic relevance feedback approach to improve the retrieval performance. Results presented on synthetic and real databases show that our approach provides better retrieval precision compared to the case when no retrieval experience is used.

1 Introduction

In interactive relevance learning approaches [1-3] for image databases, a retrieval system dynamically adapts and updates the relevance of the images to be retrieved. In these systems, images are generally represented by numeric features or attributes, such as texture, color and shape, which are called low-level visual features [4]. What user desires are called human high-level concepts. The task of relevance feedback learning is to reduce the gap between low-level visual features and human high-level concepts.

The most important thing to be learned in relevance feedback learning are the weights of different features. Learning a user's ideal query is also important. These systems deal with the situation when a single user interacts with the system for only one time. The system adapts to the user but all this experience is lost once the user terminates his/her session. In this scenario, there is only adaptation and no long-term learning. In practical applications, we desire good retrieval performance not for a single user, but for many users. Here, good retrieval performance means high precision and fast response. Although different people may associate the same image into different categories, the generalization of viewpoints of many people count much for making this decision and it will help in indexing large databases.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 102–116, 2001.

At the very beginning, images in the database have no high-level conceptual information. With more and more users performing retrieval tasks, based on their feedback, it is possible for the system to capture this experience and learn image class distribution in the sense of high-level concepts obtained during the earlier experience of the system retrieval. This method can give better results than those which are purely based on low-level features since we have extra knowledge of high-level classification. This information can significantly improve the performance of the system that includes both the instantaneous performance and the performance at each iteration of relevance feedback.

The above discussion raises two fundamental questions: (A). How to learn class distribution in the sense of high-level concepts from different users' queries and associated retrievals? (B). How to develop a better relevance learning method by integrating low-level features and high-level class distribution knowledge?

The key contribution of the paper is to present a new approach to address both of these questions. Based on the semi-supervised fuzzy c-means (SSFCM) clustering [5, 6], we propose a modified fuzzy clustering method which can effectively learn class distribution (meta knowledge) in the sense of high-level concept from retrieval experience. Using fuzzy rules, we incorporate the meta knowledge into the relevance feedback method to improve the retrieval performance. Meta knowledge consists of a variety of knowledge extracted from prior experience of the system. In this paper, we limit ourselves to the class specific information.

This paper is organized as follows. Section 2 describes the related research on learning visual concepts. Section 3 gives our technical approach for improving retrieval performance by incorporating meta knowledge into relevance feedback method. Experimental results are provided in Section 4 and Section 5 presents the conclusions of the paper.

2 Related Research

Since there is a big gap between high-level concepts and low-level image features, it is difficult to extract semantic concepts from low-level features. Recently, Chang *et al.* [7] proposed the idea of semantic visual templates (*SVT*), where templates represent a personalized view of a concept. The system interacting with the user generates a set of queries to represent the concept. However, the system does not accommodate multiple concepts which may be present in a single image and their interactions. Naphade *et al*. used the concept of multijects (probabilistic multimedia objects) for indexing which can handle queries at the semantic level. This approach can detect events such as "*explosion*" and "*waterfall*" from video. Lim [9] proposed the notion of visual keywords for content-based retrieval, which can be adapted to visual content domain via learning from examples generated by human during off-line. The keywords of a given visual content domain are visual entities used by the system. In this non-incremental approach no relevance feedback is used. Ratan *et al.* [10] adopted the multiple instance-learning paradigm using the diverse density algorithm as a way of modeling the ambiguity in images and to learn visual concepts. This method re-

quires image segmentation, which leads to additional preprocessing and the brittleness of the method.

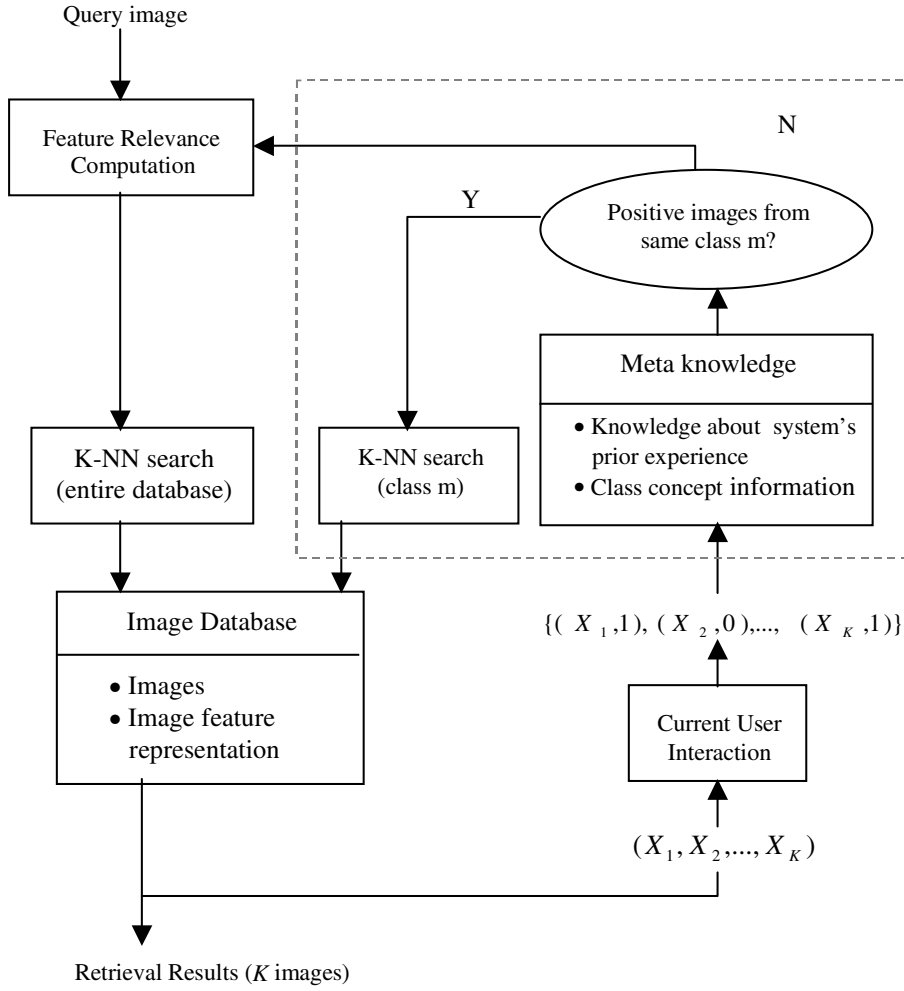


Fig. 1. System diagram for concept learning using fuzzy clustering and relevance feedback.

3 Technical Approach

Fig. 1 describes our technical approach for integrating relevance feedback with class distribution knowledge. The focus of this paper is the upper-right (dotted) rectangle.

The rest of the components shown in the figure represent a typical probabilistic feature relevance learning system.

3.1. Problem Formulation

Assume each image corresponds to a pattern in the feature space $\mathbf{R}^n$. The set of all the patterns is $\boldsymbol{X}$. We also assume the number of high-level classes c is known. After the image database (size N) has already experienced some retrievals by different users, we have $\boldsymbol{X} = \boldsymbol{X}^u \cup \boldsymbol{X}^p \cup \boldsymbol{X}^n$,where $\boldsymbol{X}^u$ represents the set of the images that are never marked (unmarked) by users in the previous retrievals; $\boldsymbol{X}^p$ represents the set of the images that are marked positive by users; $\boldsymbol{X}^n$ represents the set of the images that are marked negative by users. Note: $\boldsymbol{X}^p \cap \boldsymbol{X}^n \neq \varnothing$. The reason is that one image may be marked positive in one retrieval while marked negative in another. Even though two or more retrievals may actually be for the same high-level concept (cluster), it is still possible that the image is marked both positive and negative since whether or not to associate an image to a specific high-level concept is subjective to different users.

We provide two matrices to represent the previous retrieval experience:

(i) Positive matrix $\mathcal{P} = [\, p_{ik} \,]_{c \times N}$: If image k is ever marked positive for the ith cluster n_p times, the element $p_{ik} = n_p$; Otherwise, $p_{ik} = 0$.

(ii) Negative matrix $\mathcal{Q} = [\, q_{ik} \,]_{c \times N}$: If image k is ever marked negative for the ith cluster n_q times, the element $q_{ik} = n_q$; Otherwise, $q_{ik} = 0$.

Our problem is how to use the retrieval experience to improve the fuzzy clustering performance, i.e., make the data partition closer to a human's high-level concept.

3.2. Fuzzy Clustering

The fuzzy clustering method [11, 12, 13] is a data analysis tool concerned with the structure of the dataset under consideration. The clustering result is represented by grades of membership of every pattern to the classes established. Unlike binary evaluation of crispy clustering, the membership grades in fuzzy clustering are evaluated within the [0, 1] interval. The necessity of fuzzy clustering lies in the reality that a pattern could be assigned to different classes (categories). The objective function method is one of the major techniques in fuzzy clustering. It usually takes the form

$$J = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{p} \left\| x_k - v_i \right\|^2 \qquad (1)$$

where x_k, $k = 1, 2, \ldots, N$ are the patterns in $\mathbf{R}^n$, $v_1, v_2, \ldots, v_c$ are prototypes of the clusters, $1 < p < \infty$, and $U = [u_{ik}]$ is a partition matrix describing clustering results whose elements satisfy two conditions: (a) $\sum_{i=1}^{c} u_{ik} = 1$, $k = 1, 2, \ldots, N$; (b) $u_{ik} \geq 0$, $i = 1, 2, \ldots, c$ and $k = 1, 2, \ldots, N$.

The task is to minimize J with respect to the partition matrix and the prototypes of the clusters, namely $\min_{v_1, v_2, \ldots, v_c, U} J$, with U satisfying conditions (a) and (b).

The distance function in (1) is the Mahalanobis distance defined as

$$d_{ik} = \|x_k - v_i\|^2 = \|x_k - v_i\|^T W \|x_k - v_i\| \tag{2}$$

where W is a symmetrical positive definite matrix in $\mathbf{R}^n \times \mathbf{R}^n$.

The fuzzy c-means (FCM) method is often frustrated by the fact that lower values of J do not necessarily lead to better partitions. This actually reflects the gap between numeric-oriented feature data and classes understood by humans. The semi-supervised FCM method attempts to overcome this limitation [5, 6, 14] when the labels of some of the data are already known.

3.2.1. Semi-supervised c-means fuzzy clustering: Pedrycz [5] modified objective function J given by (1) as

$$J_1 = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik} - f_{ik} b_k)^2 d_{ik}^2 \tag{3}$$

where $b_k = 1$ if x_k is labeled, and $b_k = 0$ otherwise, $k = 1, 2, \ldots, N$. The matrix $F = [f_{ik}]_{cxN}$ with the given label vectors in appropriate columns and zero vectors elsewhere. α ($\alpha \geq 0$) denotes a scaling factor whose role is to maintain a balance between the supervised and unsupervised component within the optimization process. α is proportional to the rate N/M where M denotes the number of labeled patterns.

The estimations of cluster centers (prototypes) and the fuzzy covariance matrices are

$$v_s = \sum_{k=1}^{N} [u_{sk}^2] x_k \Big/ \sum_{k=1}^{N} [u_{sk}^2] \tag{4}$$

and

$$W_s^{-1} = \left[\frac{1}{\rho_s \det(P_s)} \right]^{1/n} P_s \tag{5}$$

respectively, where $s = 1, 2, \ldots, c$, $\rho_s = 1$ (all clusters have the same size), and

$$P_s = \frac{\sum_{k=1}^{N} u_{sk}^2 (x_k - v_s)(x_k - v_s)^T}{\sum_{k=1}^{N} u_{sk}^2}, s = 1, 2, \cdots, c \tag{6}$$

The Lagrange multiplier technique yields an expression for partition matrix

$$u_{st} = \frac{1}{1+\alpha}\left\{\frac{1+\alpha\left(1-\sum_{j=1}^{c} f_{jt}\right)}{\sum_{j=1}^{c}\frac{d_{st}^2}{d_{jt}^2}} + \alpha\left(f_{st} b_t\right)\right\} \quad s = 1,2,\cdots,c, \quad t = 1,2,\cdots,N \tag{7}$$

Using an alternating optimization (AO) method, the SSFCM algorithm iteratively updates the cluster centers, the fuzzy covariance matrices and the partition matrix by (4), (5) and (7), respectively until some termination criteria are satisfied.

3.2.2. Proposed semi-supervised fuzzy clustering method for class distribution learning: We first pre-process the retrieval experience using the following rules (i = 1, 2, …, c *and* k = 1, 2, …, N)

(i) If $p_{ik} >> q_{ik}$, we can conclude that image k should be ascribed into the ith cluster, i.e., u_{ik} should be large compared to other u_{jk} (j = 1, 2, …, c, $j \neq i$);

(ii) If $p_{ik} << q_{ik}$, we can conclude that image k should NOT be ascribed into the ith cluster, i.e., u_{ik} should be close to zero;

(iii) If (i) and (ii) are not satisfied, we cannot make any conclusion on ascribing image k (k = 1, 2, …, N) into the ith cluster, i.e., we have no idea on the value of u_{ik} so we have to execute fuzzy clustering to derive its value.

Following the above discussion, we construct two new matrixes $P_{c\times N}$ and $Q_{c\times N}$, the first of which represents positive information while the latter represents the negative information. For element p_{ik} of P, if p_{ik} and q_{ik} satisfy (i), p_{ik} = 1; Otherwise, p_{ik} = 0. For element q_{ik} of Q, if p_{ik} and q_{ik} satisfy (ii), q_{ik} = 1; otherwise, q_{ik} = 0.

We then normalize non-zero columns of P, namely, if $\sum_{i=1}^{c} p_{ik} > 0$, then $p_{jk} = p_{jk} \Big/ \sum_{i=1}^{c} p_{ik}$, j = 1, 2, …, c, k = 1, 2, …, N. The purpose of normalization is to estimate the membership grades of the marked images.

Our objective function is similar to that in (3) with the modification

$$J_2 = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N} (u_{ik} - p_{ik})^2 d_{ik}^2 \tag{8}$$

The task is to minimize the objective function J_2 with respect to the partition matrix and the prototypes of the clusters, namely $\min_{v_1, v_2, \ldots, v_c, U} J_2$ with respect to cluster centers v_1, v_2, …, v_c and U satisfying conditions (a) and (b) for the fuzzy clustering and a new constraint: u_{ik} = 0 if q_{ik} = 1, i = 1, 2, …, c, k = 1, 2, …, N. This new constraint implies that if we already definitely know that a pattern should not be ascribed to a certain class, we can pre-define the corresponding membership element to be zero. For the kth column of $Q_{c\times N}$, there are $n(k)$ non-zero elements, whose row indices are $I(k) = \{r_{1,k}, r_{2,k}, \ldots, r_{n(k),k}\}$. All other notations are the same as those in the first part of this section.

Using the technique of Lagrange multipliers, the optimization problem in (8) with constraints (a) and (b) for the fuzzy clustering, it is converted into the form of unconstrained minimization

$$J_2 = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2 d_{ik}^2 + \alpha \sum_{i=1}^{c}\sum_{k=1}^{N} (u_{ik} - p_{ik})^2 d_{ik}^2 - \sum_{k=1}^{N} \lambda_k \left(\sum_{i=1}^{c} u_{ik} - 1 \right) \tag{9}$$

From the optimization requirement $\frac{\partial J_2}{\partial u_{st}} = 0$, we get $u_{st} = \frac{1}{1+\alpha}\left\{ \frac{\lambda_t}{2d_{st}^2} + \alpha p_{st} \right\}$, if q_{st} = 0; otherwise, $u_{st} = 0$.

From the fact that the sum of the membership values, $\sum_{j=1}^{c} u_{jt} = 1$, we have $\frac{1}{1+\alpha}\left\{ \frac{\lambda_t}{2} \sum_{j=1, j\notin I(t)}^{c} \frac{1}{d_{jt}^2} + \alpha \sum_{j=1, j\notin I(t)}^{c} p_{jt} \right\} = 1$. So we can derive

$$u_{st} = \frac{1}{1+\alpha}\left\{ \frac{1+\alpha-\alpha \sum_{j=1, j\notin I(t)}^{c} p_{jt}}{\sum_{j=1, j\notin I(t)}^{c} \frac{d_{st}^2}{d_{jt}^2}} + \alpha p_{st} \right\} \tag{10}$$

The expressions of cluster centers and the fuzzy covariance matrices are the same as in (4) and (5) respectively. Our semi-supervised fuzzy clustering algorithm for learning class distribution is outlined in Fig. 2.

1. Given the number of clusters c, positive matrix P, negative matrix Q. Select the distance function as Euclidean distance.
2. Compute new matrixes $P_{c\times N}$ and $Q_{c\times N}$. Initialize partition matrix U: If $q_{ik} = 1$, $u_{ik} = 0$; Otherwise, set u_{ik} randomly in the interval [0, 1] so that the sum of each column of U is 1.
3. Compute cluster centers and the fuzzy covariance matrices by (4) and (5).
4. Update partition matrix: If $q_{ik} = 1$, $u_{ik} = 0$; Otherwise, compute the element by (10).
5. If $\|U - U'\| < \delta$ (with δ being a tolerance limit) then stop, else go to 3 with $U = U'$.

Fig. 2. Semi-supervised fuzzy clustering algorithm (SSFCM) for class distribution learning.

3.3. Incorporating Meta Knowledge into Feature Relevance Learning

A kind of probabilistic feature relevance learning (PFRL) based on user's feedback, that is highly adaptive to query locations is suggested in [3]. The main idea is that feature weights are derived from probabilistic feature relevance on a specific query (local dependence), but weights are associated with features only. Fig. 3 illustrates the cases at points near decision boundary where the nearest neighbor region is elongated in the direction parallel to decision boundary and shrunk in the direction orthogonal to boundary. This implies that the feature with direction orthogonal to the decision boundary is more important. This idea is actually the adaptive version of nearest neighbor technique developed in [15].

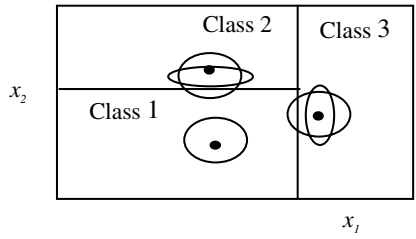


Fig. 3. Feature weights are different along different dimensions. The dotted circles represent the equally likely nearest neighborhood and the solid ellipses represent feature-weighted nearest neighborhood.

3.3.1. Proposed strategy for relevance feedback with fuzzy clustering: Fig. 1 describes our strategy for relevance feedback using class distribution knowledge. If we ignore the components in the upper-right (dotted) rectangle, the remaining represents a typical probabilistic feature relevance learning system. We now introduce the three components in the upper-right rectangle.

Using fuzzy clustering, we already get class distribution knowledge, which is represented by the partition matrix $U_{c\times N}$. We now de-fuzzy this meta knowledge, i.e., update the elements of U by binary scale {0, 1}. The de-fuzzy rule is: If $u_{ik} \geq \beta(\max_{j=1,2,\ldots,c} u_{jk})$, u_{ik} =1; else, u_{ik} = 0, i = 1, 2, …, c, k = 1, 2, …, N. The value of $\beta \in (0, 1]$ represents to what extent we can say that the element u_{ik} is large enough so that image k can be ascribed to class i. Notice that this concept learning is not incremental.

At any iteration, if M images (I_1, I_2, …, I_M) are marked positive by the current user, we then check if these positive images can be ascribed into one common class. If $\exists s \in \{1, 2, \ldots, c\}$, $\forall$ k $\in$ { I_1, I_2, …, I_M} that u_{sk} = 1, then the current user seems to be seeking the concept corresponding to class s. So the system can save the tremendous amount of work for feature relevance learning and searching K images over the entire database; Instead, only searching K images within class s is needed, i.e., searching among the images whose sth element of the corresponding U column vectors are 1.

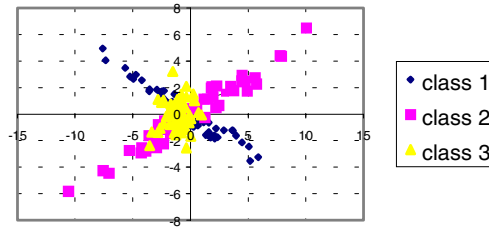


Fig. 4. Two-dimensional data distribution with three overlapping clusters.

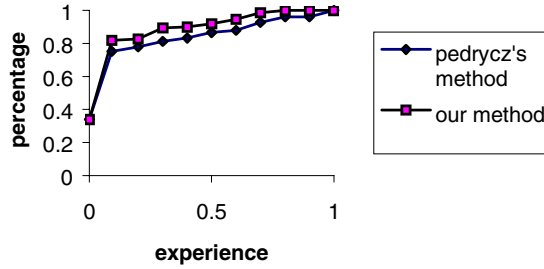


Fig. 5. Clustering results by two methods with different experience.

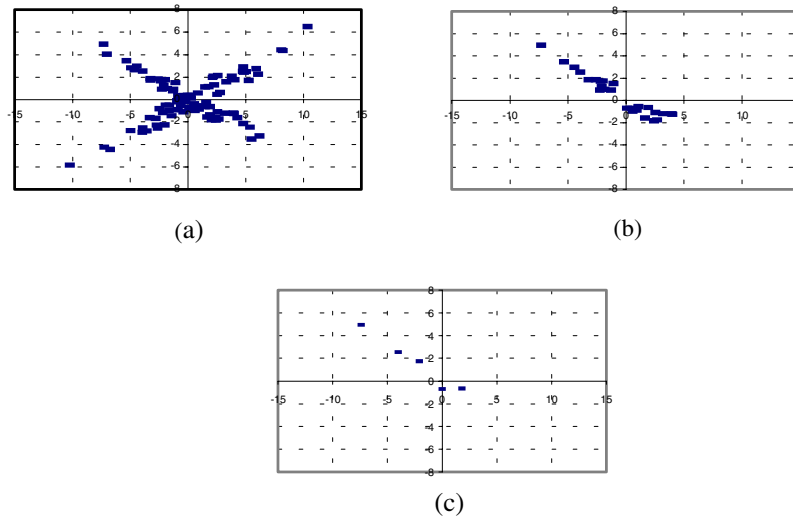


Fig. 6. Misclassified patterns for two-dimensional data set: (a) no experience, (b) 20% experience, (c) 50% experience.

When enough retrievals on the image database are executed by different users, the class distribution knowledge will be close to most human users concepts. This leads to not only saving computational time for retrieval, but also to improved precision for retrieval.

Fig. 7. Sample Images from real-world database: (a) images having one concept; (b) images having two concepts; (c) images having three concepts; (d) images having four concepts.

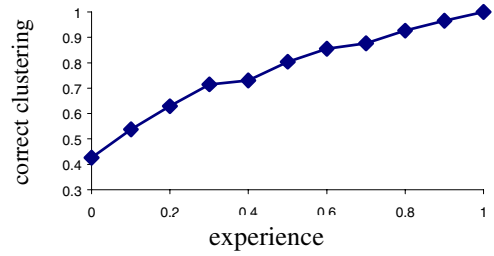


Fig. 8. Clustering results for real data with different experience.

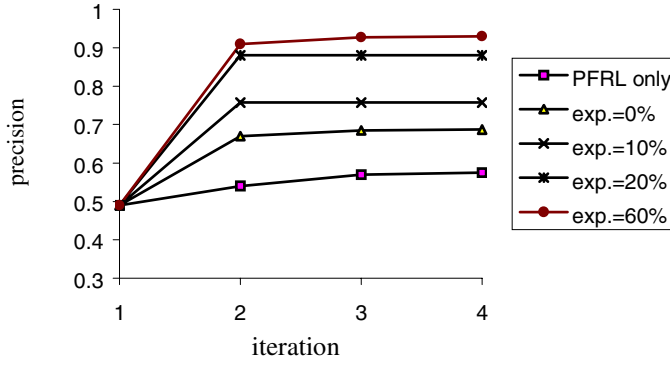


Fig. 9. Retrieval precisions for different experience.

4 Experiments

4.1. Synthetic Data

Fig. 4 shows a synthetically created two-dimensional pattern. It consists of three overlapping clusters: two of them are ellipsoidal (class 1 and class 2) while the third one (class 3) is a circle. The two ellipsoidal clusters have the same means $[\ 0\ 0\ ]^T$, and their covariance matrix given as rows are [12 -6.8 ; -6.8 4] and [12 6.8 ; 6.8 4] respectively. The third cluster has mean of $[\ -1\ 0]^T$ and its covariance matrix is [1 0 ; 0 1]. The size of each cluster is 50, so we have 150 patterns in total. For standard fuzzy clustering, the correct percentage is only 36.7%, which is close to the guess value 1/3. This is not unusual because clusters significantly overlap.

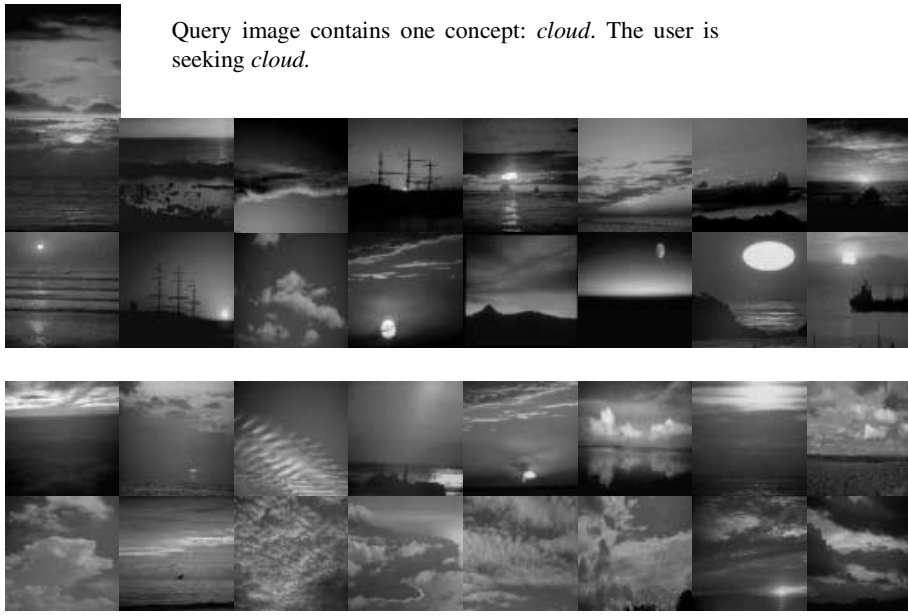
We then test both Pedrycz's clustering algorithm [5] and our algorithm on this data with different amounts of experience. Experience is defined as the ratio of the number of labeled patterns to the total number of patterns. When the experience is γ, we randomly choose γN patterns and label them positive for their groundtruth clusters; at the same time, randomly choose γN patterns, and for each pattern, label it negative for one cluster that is not its ground-truth cluster. Then repeat clustering with respect to this experience 10 times, and calculate the average correct percentage. For Pedrycz's method, only positive experience is used while for our method both positive and negative experiences are used. Fig. 5 shows that with increasing experience, the clustering results become better and that the results of our method are better than Pedrycz's. Fig. 6 shows the misclassified patterns by our method with respect to different experience values. This shows the advantage of our algorithm for learning high level concepts since in addition to positive feedback, negative feedback is also available from user's responses.

4.2. Real Data

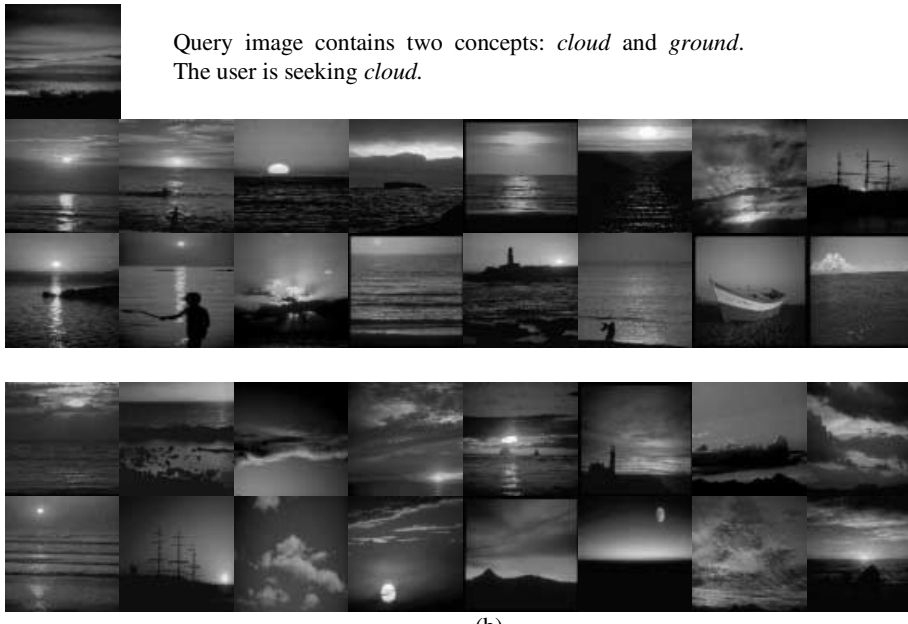
We construct a real-world image database, consisting of a variety of images all containing one or more of the following five objects: water, sun, sky, cloud and ground. The total number of images is 180. Each image is annotated with five labels (0 or 1), so the groundtruth class distribution can be represented by a matrix $G_{180\times5}$ whose elements are 0-1 value. Fig. 7 shows sample images. The numbers of images within the five classes are 49, 63, 83, 130, 59, respectively. Each image in this database is represented by 16-dimensional feature vectors obtained using 16 Gabor filters for feature extraction [2].

Our semi-supervised fuzzy clustering algorithm is applied to the data with different amounts of experience, $N = 180$, $c = 5$, $\alpha = 1$, $\beta = 0.5$. Fig. 8 shows the correct clustering percentage with respect to different experience. The clustering correctness is determined by comparing the elements of the ground-truth matrix G and those of defuzzied partition matrix U.

We then randomly select one of the 180 images as query, and other 179 remaining images as training samples. The retrieval process is automatically executed since we

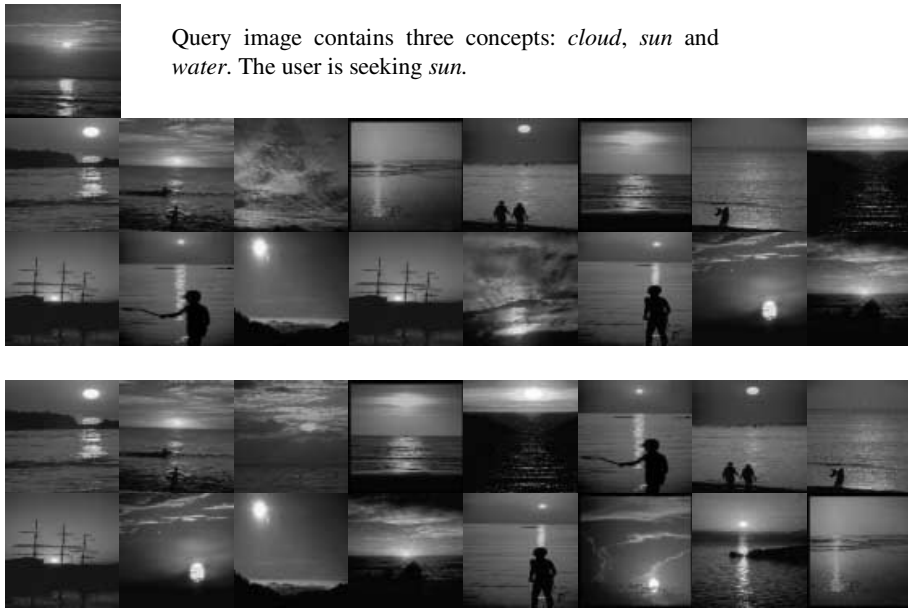


(a)

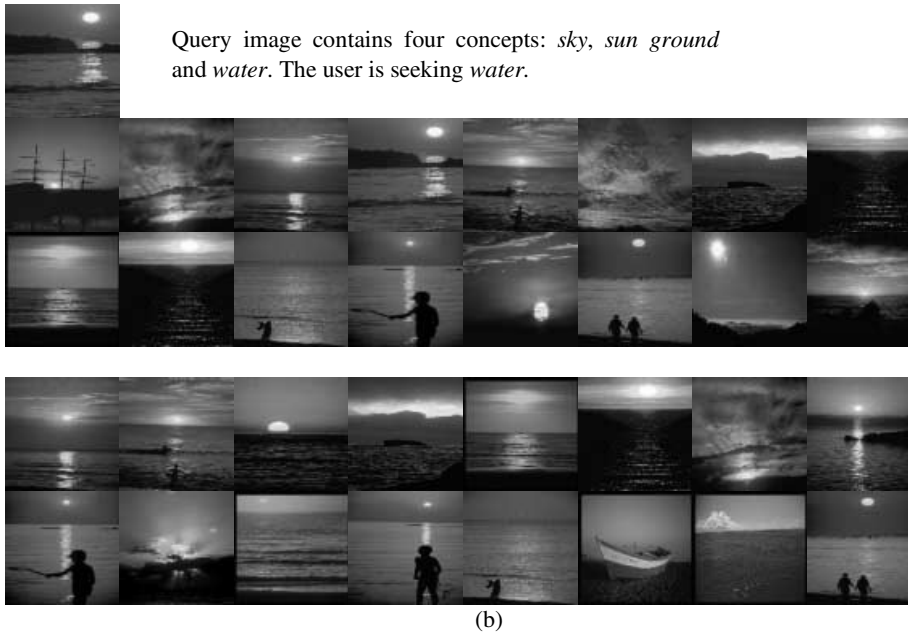


(b)

Fig. 10. The sample (top 16) retrieval results (experience = 20%) at the first and the second iterations with query image containing (a) one concept, (b) two concepts.



(a)



(b)

Fig. 11. The sample (top 16) retrieval results (experience = 20%) at the first and the second iterations with query image containing (a) three concepts, (b) four concepts.

use the groundtruth matrix $G_{180\times5}$ to provide user's interactions: At first, randomly select a concept that the query image can be ascribed to, and regard this concept as what the user is seeking. When the retrieval system presents the resulting K images, we use matrix $G_{180\times5}$ to mark them. If the membership element of the $G_{180\times5}$ corresponding to the image with respect to desired concept is 1, then mark this image positive; otherwise, it is marked as negative. By repeating such retrievals 50 times by selecting a different image as query each time, we obtain the average results shown in Fig. 9.

The performance is measured using the average retrieval precision defined as

$$\text{precision} = \frac{\text{Number of total retrievals}}{\text{Number of positive retrievals}} \times 100\%$$

We observe that when only PFRL is used, the average precision (= 58.1%) is the lowest. With the increasing experience, the average precision becomes higher. Experience of 10% helps to increase the precision significantly (precision = 68.9%). When the experience is 20%, the precision reaches 88.0%. These results support the efficacy of our method.

Fig. 10 and Fig. 11 show four groups of sample retrievals in total when 20% experience is available. The query image in each group contains different number of concepts from 1 to 4. The retrieval results at the second iterations are improved over those at the first iterations with the help of meta knowledge derived from the experience using fuzzy clustering. For example, the query image in Fig. 10. (b) contains two concepts: *cloud* and *ground*. The user is seeking the concept *cloud.* At the first iteration, the system makes K-nearest neighbor search and only 5 out of the 16 resulting images contain *cloud*. At the second iteration, the system incorporates the class distribution knowledge into relevance feedback framework and 14 out of 16 images contain *cloud*.

5 Conclusions

This paper presented an approach for incorporating meta knowledge into the relevance feedback framework to improve image retrieval performance. The modified semi- supervised fuzzy clustering method can effectively learn class distribution in the sense of high-level concept from retrieval experience. Using fuzzy rules, we adapted the meta knowledge into relevance feedback to improve the retrieval performance. With more retrievals on the image database by different users, the class distribution knowledge became closer to typical human concepts. This leads faster retrieval with improved precision. The consequence of this is to be able to handle more effectively a large database. In the future, we will show results on a larger and more complex image database.

Acknowledgements

This work was supported by DARPA/AFOSR grant F49620-97-1-0184. The contents of the information do not necessarily reflect the position or the policy of the US Government.

References

1. Y. Rui, T. Huang, M. Ortega and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," *IEEE Trans. on Circuits and Systems for Video Technology,* pp. 644-655, Vol. 8, No. 5, September 1998.
2. J. Peng, B. Bhanu, and S. Qing, "Probabilistic feature relevance learning for content-based image retrieval," *Computer Vision and Image Understanding*, Vol. 75, pp. 150 - 164, July/August, 1999.
3. T. Minka and R. Picard, "Interactive learning with a society of models," *Pattern Recognition,* Vol. 30, No. 4, pp. 565-581, 1997.
4. M. Flickner et al., "Query by image and video content: the QBIC system," *IEEE Computer,* pp. 23-31, September, 1995.
5. W. Pedrycz and J. Waletzky, "Fuzzy clustering with partial supervision," *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. 27, No. 5, pp. 787-795, October, 1997.
6. W. Pedrycz, "Algorithm of fuzzy clustering with partial supervision", *Pattern Recognition Letters,* Vol. 3, pp. 13-20, January, 1985.
7. S.F. Chang, W. Chen, and H. Sundrarm, "Semantic visual templates - linking features to semantics," *Proc. 5th IEEE International Conference on Image Processing*, Vol. 3, pp. 531-535, Chicago, IL, October 1998.
8. M. R. Naphade, T. Kristjansson, B. Frey, T. S. Huang, *Probabilistic Multimedia Objects (Multijects): a Novel Approach to Video Indexing and Retrieval in Multimedia Systems*, *Proc. International Conference on Image Processing*, Vol. 3, pp. 536-540, 1998.
9. J.H. Lim, "Learning visual keywords for content-based retrieval," *Proc. IEEE International Conference on Multimedia Computing and Systems*, Vol. 2, pp. 169-173, 1999.
10. A.L. Ratan, O. Maron, W.E.L. Grimson, "A Framework for learning query concepts in image classification," *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 423-429, 1999.
11. A. Jain, M. Murty, and P. Flynn, "Data clustering: a review," *ACM Computing Surveys*, Vol. 31, No. 3, September, 1999.
12. J. C. Bezdek, J. Keller, R. Krisnapuram, and N. R. Pal, *Fuzzy Models and Algorithms for Pattern Recognition and Image Processing*. Kluwer Academic Publisher, 1999.
13. D. E. Gustafson and W. C. Kessel, "Fuzzy clustering with a fuzzy covariance matrix," in *Proc. IEEE Conference on Decision and Control*, pp. 761-766, 1978.
14. M. Bensaid, L. O. Hall, J. C. Bezdek, and L. P. Clarke, "Partially supervised clustering for image segmentation," *Pattern Recognition*, Vol. 29, No. 5, pp 859-871, 1996.
15. T. Hastie and R. Tibshirani, "Discriminant adaptive nearest neighbor classification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 18, No. 6, pp. 607-616, June, 1996.

LC: A Conceptual Clustering Algorithm

Jos Fco. Mart nez-Trinidad[1] and Guillermo S nchez-D az[2]

[1]Centro de Investigaci n en Computaci n, IPN
Juan de Dios Batiz s/n esq. Oth n de Mendizabal, UPALM; C.P. 07738, M xico D.F.
fmartine@cic.ipn.mx
[2]Centro de Investigaci n en Tecnolog as de Informaci n y Sistemas, UAEH
Carretera Pachuca-Tulancingo, Km. 4.5, C.P. 42074, Pachuca, Hidalgo, M xico.
sanchezg@uaeh.reduaeh.mx

Abstract. An algorithm for automated construction of conceptual classification is presented. The algorithm arranges the objects into clusters using clustering criterion based on graph theory and constructs the concepts based on attributes that distinguish objects in the different computed clusters (typical testors). The algorithm may be applied in problems where the objects can be described simultaneously by qualitative and quantitative attributes (mixed data); with incomplete descriptions (missing data) and the number of clusters to be formed is not known a priori. LC has been tested on data from standard public databases.

1. Introduction

The conceptual clustering algorithms surge with the researches of R.S. Michalski at 80 s [1]. In these works the main objective was to give additional information that classical unsupervised techniques of Pattern Recognition do not give. The classical techniques only build clusters, while the conceptual clustering algorithms build clusters and explains (through concepts or properties) why a set of objects conform a cluster.

Algorithms as Cobweb [2], Unimem [3], Witt [4], Linneo$^+$ [5], Conceptual K-means [6], were proposed in order to solve conceptual clustering.

In general, we can observe that for handling mixed qualitative and quantitative data, the proposed conceptual algorithms (incremental and non-incremental) attempt to do the following:

a) Code *qualitative attribute values as quantitative values, and apply distance measures used in quantitative situations*. The transformation of qualitative information in quantitative information in order to compute any arithmetical operations on this last information does not make sense, and the resultant similarity values are difficult to interpret.

b) Transform *numeric attributes into categorical attributes and apply an algorithm that handles only qualitative information*. To do that, a certain categorization process is needed. Often this process causes loss of important information especially the relative (or absolute) difference between values for the numeric attributes. Besides,

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 117 127, 2001.

the original problem must be modified or the representation space is changed. This sometimes implies loss of clarity and therefore, loss of trustworthiness.

c) *Generalize comparison functions designed for quantitative attributes to handle* quantitative *and qualitative attribute values*. The functions used for quantitative attributes are based on distances, which cannot be extended to handle qualitative also, because both are in different spaces. Several attempts violate this condition by evaluating the total distance as the addition of the distance between qualitative attributes and the distance between quantitative attributes. Moreover, consider that the result is in the original n-dimensional space, where the centroid can be calculated.

On the other hand, the concepts that built all the algorithms before mentioned are statistical descriptors; these kinds of descriptors are hard to interpret by final users, who usually are not specialists in statistics.

The proposed algorithm in this paper is based on unsupervised classification concepts of Pattern Recognition. In addition, it takes some ideas proposed by Michalski in order to build interpretable concepts (logical properties) using attributes, which are used to describe the objects in the sample under study. The LC algorithm solves conceptual clustering problems where the objects are described by qualitative and quantitative attributes, where may be present missing values, and the concepts built by the algorithm are not statistical properties of the clusters.

2. LC Conceptual Algorithm

It is known from the Set Theory that any set can be intensionally or extensionally determined. The unsupervised conceptual classification problems reflect this double situation. Therefore, our proposed algorithm follows this idea.

The LC-conceptual algorithm consists of two stages. The first, denominated *extensional structuralization*, it is where the clusters are constructed. With this purpose clustering criteria based on similarity measure between objects are used. The second stage, denominated *intensional structuralization*, it is where the concepts associated with the clusters are built. For this task, sets of appropriate attributes for constructing the associated concepts to each cluster are selected. In this stage, we introduce a genetic algorithm instead of the use of a traditional algorithm (which has exponential run time complexity) to compute these sets of attributes.

2.1 Extensional Structuralization

In the extensional structuralization step, the goal is to find clusters of objects. Therefore, we will use the unsupervised logical combinatorial pattern recognition concepts (see [7,8]). In this context, an unsupervised problem is expressed as follows:

Let Ω a universe of objects and $MI=\{I(O_1),..., I(O_m)\}$ be an object description set, where $O_i \in \Omega$ $i=1,...,m$. A description $I(O)$ is defined for every object O by a finite sequence $x_1(O),\ldots,x_n(O)$ of values associated with n attributes of the set $\Re=\{x_1,\ldots,x_n\}$, where $x_i(O)\in D_i$, and D_i is the set of all admissible values for attribute x_i. Additionally, we will assume that in D_i $(i=1,\ldots,n)$ there exists a symbol "?" which

denotes *absence of information* (*missing data*). In other words, an object description could be incomplete, i.e., there is at least one variable in at least one object for which we do not know its value. We will consider that $I(O) \in D_1 \times \square \times D_n$=IRS (Initial Representation Space, the Cartesian product of admissible value sets of each attribute). The types of these attributes are not necessarily the same. For example, some of them could be qualitative (i.e. Boolean, many-valued, fuzzy, linguistic, etc.) and others, quantitative (i.e. integer, real, interval, etc.), so we do not restrict IRS to have any algebraic or topologic structure. We do not restrict M_i either to have any *a priori* defined algebraic or logic operations or any distance (metric).

Let Γ: IRS × IRS $\rightarrow L$ a function, where L is a totally ordered set; Γ will be denominated *similarity function* and it is an evaluation of the similarity degree between any two descriptions of objects belonging to Ω. Any restriction of Γ to any subset $T \subseteq \Re$ we will be called *partial similarity function*. Often we will consider functions that do not satisfy the properties of a metric. In general, we can consider functions that are non-positive-definite, do not fulfill the triangular inequality, and L is not a subset of the real numbers. In other words, in principle, IRS is a simple Cartesian product. Usually, this information about the objects (their descriptions) is given in the form of a table or matrix $MI=\left| x_i(O_j) \right|_{mxn}$ with m rows (object descriptions) and n columns (values of each attribute in the selected objects).

The problem of the extensional structuralization consists of the determination of the covering set $\{K_1, \square, K_c\}$, $c>1$, of Ω. This set could be a partition or simply a cover.

We will use clustering criteria based on topological relationships between objects. This approach responds to the following idea: given a set of object descriptions, find or generate *natural* clusters of these objects in the representation space IRS. This structuralization must be achieved using some similarity measure between objects based on a certain property. In practice, this property reflects the relationship between objects according to a model given by the expert in a concrete area.

From now on, we will use O instead of $I(O)$ to simplify the notation. A clustering criteria have as parameters a symmetric matrix (if Γ is a symmetric function) $\left| \Gamma_{ij} \right|_{mxm}$, denominated *similarity matrix,* in which each $\Gamma_{ij}=\Gamma(O_i,O_j) \in L$; a property Π which establishes the way we can use Γ; and a *threshold* $\beta_0 \in L$. We say that O_i, $O_j \in MI$ are β_0*-similar* iff $\Gamma(O_i,O_j) \geq \beta_0$. In the same way, we say that $O_j \in MI$ is a β_0*-isolated element* iff $\forall O_j \neq O_i \in MI \ \Gamma(O_i,O_j) < \beta_0$. Thus, clusters are determined by imposing the fulfillment of properties over the similarities between objects (clustering criterion) [9].

Note that there exist a natural correspondence between IRS and a graph whose vertexes are object descriptions, and the weight of their edges is the β_0-similarity between adjacent vertexes. The value β_0 is a user-defined parameter that can be used to control how similar a pair of objects must be in order to be considered similar. Depending on the desired closeness in similarity, an appropriate value of β_0 may be chosen by the user.

Definition. For a *crisp clustering criterion* $\Pi(MI, \Gamma, \beta_0)$ [9] we mean a set of propositions with parameters MI, Γ and β_0 such that:

it generates a family $\tau=\{K_1,....,K_c\}$ of subsets of MI (*crisp clusters*) that:

i)$\forall K_i \in \tau \, [K_i \neq \varnothing]$;

ii) $\bigcup_{K_i \in \tau} K_i = MI$;

iii) $\neg\exists\, K_r, K_{j_1}, ..., K_{j_p} \in \tau\, [K_r \subseteq \bigcup_{\substack{t=1 \\ j_t \neq r}}^{p} K_{j_t}]$;

and it defines a relation $R_\Pi \subseteq MI \times MI \times 2^{MI}$ (where 2^{MI} denotes the power set of MI) such that:

iv) $\forall O_i, O_j \in MI\, [\exists K_i \in \tau\, \exists\, S \subseteq MI\, [O_i, O_j \in K_i \Leftrightarrow (O_i, O_j, S) \in R_\Pi]$

The family τ gives the final cover. Therefore, τ represents the extensional structuralization of the sample MI.

Examples of clustering criteria are the following:

Definition. We say that a subset $K_r \neq \varnothing$ of MI is a β_0-*connected cluster* [10] iff:

a) $\forall O_i, O_j \in K_r\, \exists\, O_{i_1}, ..., O_{i_q} \in K_r\, [O_i = O_{i_1} \wedge O_j = O_{i_q} \wedge \forall p \in \{1, ..., q-1\}\ \Gamma(O_{i_p}, O_{i_{p+1}}) \geq \beta_0]$

b) $\forall O_i \in MI\, [(O_j \in K_r \wedge \Gamma(O_i, O_j) \geq \beta_0) \Rightarrow O_i \in K_r]$

c) Any β_0-isolated element is a β_0-*connected cluster* (*degenerated*).

Definition. We say that a subset $K_r \neq \varnothing$ of MI is a β_0-*compact cluster* [10] iff:

a) $\forall O_j \in MI$

$[O_i \in K_r \wedge (\max_{\substack{O_t \in MI \\ O_t \neq O_i}} \{\Gamma(O_i, O_t)\} = \Gamma(O_i, O_j) \geq \beta_0 \vee \max_{\substack{O_t \in MI \\ O_t \neq O_j}} \{\Gamma(O_j, O_t)\} = \Gamma(O_j, O_i) \geq \beta_0)] \Rightarrow O_j \in K_r$

b) $\forall O_i, O_j \in K_r\, \exists\, O_{i_1}, ..., O_{i_q} \in K_r\, [O_i = O_{i_1} \wedge O_j = O_{i_q} \wedge \forall p \in \{1, ..., q-1\}$

$[\max_{\substack{O_t \in MI \\ O_t \neq O_{i_p}}} \{\Gamma(O_{i_p}, O_t)\} = \Gamma(O_{i_p}, O_{i_{p+1}}) \geq \beta_0 \vee \max_{\substack{O_t \in MI \\ O_t \neq O_{i_{p+1}}}} \{\Gamma(O_{i_{p+1}}, O_t)\} = \Gamma(O_{i_p}, O_{i_{p+1}}) \geq \beta_0]]$

c) Any β_0-isolated element is a β_0-*compact cluster* (*degenerated*).

In [10] other clustering criteria are proposed and in addition, relations of inclusion among the clusters generated by these criteria are proved.

Note that after applying a clustering criterion we can know the extension or list of objects that constitute each cluster.

2.2 Intensional Structuralization

In the intensional structuralization step, the property (concept) that satisfies each cluster is built. A relational proposition $[x_i \# R_i]$ where $R_i \subseteq M_i$, and # symbolize the relational operators $\geq$, $>$, $<$, $\leq$, $=$, $\in$ or their negations, is called a *selector* (see [1]). The selector $[x_i = R_i]$ ($[x_i \neq R_i]$) is interpreted as "the value of $x_i \in R_i$"(the value of $x_i \notin R_i$ "). A logical product of selectors is called a *logical complex* (*l-complex*, see [1]).

Definition. The *REFUNION on t* operator is a function $RU_t: 2^{\Omega} \cup 2^{L(\Omega)} \rightarrow L(\Omega)$; where t is the set of attributes used to define the l-complexes; and $L_t(\Omega)$ is the set of all l-complexes under Ω defined in terms of the attributes in t. It transforms a set of objects and/or l-complexes defined in terms of the attributes in t, in an l-complex defined in terms of the same set of attributes.

Before the intensional structuralization step, we have already obtained $K_1, ..., K_c$ clusters in the extensional structuralization stage. In the intensional structuralization

step, we will use the concept of typical testor for $K_1,...,K_c$ [7,11,12]. A *testor* is a subset of attributes $t=\{x_{i_1},...,x_{i_t}\}$ such that if we consider only these attributes, similar objects in different clusters then do not appear. A *typical testor* is a testor for which none of its proper subsets is a testor.

In this paper we will use typical testors of a cluster with respect to the complement (*typical testors by class*, see [13]). Typical testors by class distinguish objects of the cluster K_i from objects in the union of the other clusters K_j $j=1,...,c. j\neq i$. Therefore, we can use them in order to construct the *l*-complexes using the *REFUNION* on *t* operator.

We propose the use of a genetic algorithm in order to get a subset of typical testors. These typical testors will have minimal length (this characteristic is very useful because in conceptual clustering is easy to interpret short concepts). The proposed genetic algorithm [14] obtains a subset (or the total set) of typical testor of minimal length, at one time considerably less than other algorithms [15]. These algorithms carry out this procedure on exponential time (with concerning the number of attributes).

The genetic algorithm does not calculate the total set of typical testors, but the computed subset is useful, since the conceptual algorithm LC could utilize this subset in order to carry out the characterization of the resulting clusters.

In general, the genetic algorithm proceeds as follow: First, it generates the initial individual population randomly, and the size (i.e. number of points) of each individual will be the number of attributes. Each point of the individuals will be 0 or 1 value. Second, individuals are evaluated in order to determine their fitness (the algorithm verify if each individual is typical testor or not). The individuals of the population are crossed between them, to generate another new. In this procedure, the attributes of the individuals with great fitness are preserved (i.e. the individuals that were typical testors). The crossover operation generates a new individual population that could replace the previous population, or this new population could be mixed with the old population in order to get populations with better fitness.

Definition. Let ζ_i a subset of typical testors of minimal length. *The star of a cluster* K_i with respect to the clusters K_j j=1,...,c, j≠i, is the set of maximal complexes under inclusion covering any object in K_i and not covering any object in K_j j=1,...,c, j≠i. It is denoted as $G_{\zeta_i}\left(K_i \Big/ \bigcup_{j=1,...,c;\, j\neq i} K_j\right)=\left\{RU_t(K_i)=\wedge_{x_i\in I}[x_i=R_i] \Big/ t\in\zeta_i, i=1,...,c\right\}$ where RU_t is the REFUNION on *t* operator.

Another operator used in the intensional phase is the generalization operator *GEN*.

Definition. The generalization operator *GEN* transforms each $[x=R_x]$ of an *l*-complex α for K_i into a more general $[x=R'_x]$ as:

1. If *x* is a variable of interval type, the *closing interval rule* is applied:
   a)Put $I=[min, max]$, where $min=\min_{R_x}\{\min_j / I_j=\lfloor \min_j, \max_j \rfloor \in R_x\}$ and
   $$\max=\max_{R_x}\{\max_j / I_j=\lfloor \min_j, \max_j \rfloor \in R_x\}$$
   b)If *I* does not cover new objects out of K_i then
   i)$R'_x=I$

Else

i)Find the set of k disjoint subintervals $H_i \subset I$, $i=1,...,k$, such that $\forall I_j \in R_x$ $\exists! l \in \{1,...,k\} I_j \subseteq H_l$ and $\forall i=1,...,k$, H_i does not cover new objects out of K_i.

ii) $R_x'=\{H_i\}$.

2. Quantitative and ordinal qualitative variables are special cases of interval type variables having the property $\forall j=1,...,|R_x|$ $min_j=max_j$.
3. If x is a set type variable

a) Put $N = \bigcup_{j=1}^{|R_x|} v_j,\ v_j \in R_x$

b) If N does not cover new objects out of K_i, then $R'_x=N$; else $R'_x=R_x$

4. If x is a structured tree type variable, the *climbing generalization rule* is applied:

a) Let p be the lowest parent node whose descendants include all the values of R_x.

b) If $p \in R_x$ or $\{p\}$ does not cover new objects out of K_i then

i) $R'_x=\{p\}$.

Else

i) Find the minimal set of values Q that generalizes to all values of R_x such that new objects out of K_i are not covered.

ii)$R'_x=Q$.

5. If x is a type structured graph variable, the similar rule to the above case taking instead of p the set of lowest parent nodes connected with all the values of R_x is applied.
6. If x is a nominal variable, any generalization rule is not applied, that is $R'_x=R_x$.
7. For all types of variables, if $R_x=M_i$ the dropping condition rule is applied.

GEN operator obtains a generalized l-complex from α.

The block diagram of the LC conceptual clustering algorithm is shown in figure 1.

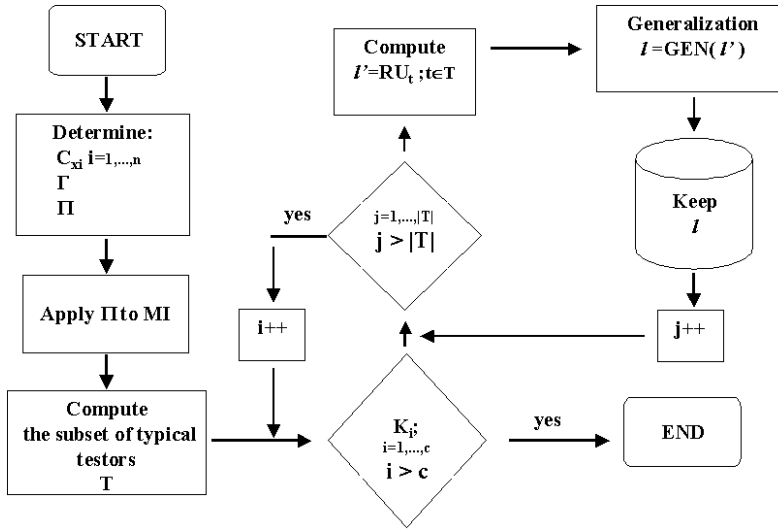


Fig. 1. Block diagram of the LC conceptual clustering algorithm

3. Experimental Results

Observe how the algorithm is applied to a data set of microcomputers. The attributes that are shown in table 1 describe each microcomputer.

Table 1. Descriptions of computers.

Microcomputer	Display	RAM	ROM	MP	Keys
Apple II	*Color TV*	*48K*	*10K*	*6502*	*52*
Atari 800	*Color TV*	*48K*	*10K*	*6502*	*57-63*
Commodore VIC20	*Color TV*	*32K*	*11K*	*6502A*	*64-73*
Exidi Sorcerer	*B&WTV*	*48K*	*4K*	*Z80*	*57-63*
Zenith H8	*Built-in*	*64K*	*1K*	*8080A*	*64-73*
Zenith H89	*Built-in*	*64K*	*8K*	*Z80*	*64-73*
HP-85	*Built-in*	*32K*	*80K*	*HP*	*92*
Horizon	*Terminal*	*64K*	*8K*	*Z80*	*57-63*
Ohio Sc. Challenger	*B&WTV*	*32K*	*10K*	*6502*	*53-56*
Ohio Sc. II Series	*B&WTV*	*48K*	*10K*	*6502C*	*53-56*
TRS-80 I	*B&WTV*	*48K*	*12K*	*Z80*	*53-56*
TRS-80 III	*Built-in*	*48K*	*14K*	*Z80*	*64-73*

Boolean comparison criteria for comparing the values of the attributes were used. Thus, for Display and Microprocessor (MP) attributes the following criterion was considered, "two values are considered similar if they are in the same set". This criterion can be formalized as $C_{Display}(x_i(O), x_i(O')) = \begin{cases} 1 & \text{if } x_i(O), x_i(O') \in \{Terminal\} \vee \\ & \text{if } x_i(O), x_i(O') \in \{B\,\&\,WTV, ColorTV\} \vee \\ & \text{if } x_i(O), x_i(O') \in \{Built-in\} \\ 0 & otherwise \end{cases}$ in the case of Display attribute.

For the rest of the attributes, the matching criterion was used. Therefore, for RAM attribute, the comparison criterion is as follows $C_{RAM}(x_i(O), x_i(O')) = \begin{cases} 1 & \text{if } x_i(O) = x_i(O') \\ 0 & otherwise \end{cases}$.

Table 2. Clusters β_0-connected with β_0=0.6.

Cluster 1		Cluster 2	Cluster 3
Atari 800	Ohio Sc. II Series	Zenith #8	HP-85
Commodore VIC20	TRS-80 I	Zenith #89	
Exidi Sorcerer	Apple II	Horizon	
Ohio Sc. Challenger		TRS-80 III	

As similarity function was used a real valued function that takes values in the interval [0,1]. It was defined as $\Gamma(O, O') = \frac{\left|\{x_i \,/\, x_i \in \Re, C(x_i(O), x_i(O')) = 1\}\right|}{|\Re|}$, where $\Re$ is the set of attributes used to describe the objects. Note that, the algorithm

allows using the criterion that the specialist in the practice handles in order to compare the values of the attributes and the descriptions of the objects.

The β_0-connected clustering criterion was used with β_0=0.6. This criterion gave us three clusters, which are shown in table 2.

The second stage in the algorithm is the construction of the concepts associated with each cluster (intensional structuralization). Following our example, it continues computing the typical testors for the clusters and the building of the concepts. In the table 3 are shown the concepts built by LC.

Table 3. Concepts for the clusters shown in the table 2 using typical testors.

Cluster 1	Cluster 2	Cluster 3
1.-ROM=[4 10 12 11-16] 2.-Display=[Color_TV B&W_TV]	1.-ROM=[1 8 14]	1.-Keys=[92] 2.-MP=[HP] 3.-ROM=[80]

These concepts explain us for example that objects in the cluster 1 are microcomputers with ROM of size 4, 10, 12 and 11-16, the objects in cluster 2 have ROM of size 1, 8 and 14. And the microcomputers in the cluster 3 have ROM of size 80. In this example all the typical testors were of length one.

Table 4. Extensional structuralization of the zoo data.

				Cluster 1				
Aardvark	calf	polecat	buffalo	hamster	cavy	wallaby	Squirrel	pussycat
Bear	cheetah	pony	deer	fruitbat	dolphin	wolf	Mongoose	lion
Girl	goat	puma	elephant	hare	porpoise	boar	Platypus	reindeer
Gorilla	leopard	raccoon	giraffe	vampire	seal	antelope	Oryx	vole
Sealion	lynx							
				Cluster 2				
Bass	herring	pitviper	dogfish	tuatara	frog	chub	Seasnake	frog
Catfish	piranha	stingray	pike	newt	toad	slowworm	Tuna	
				Cluster 3				
Carp	haddock	seahorse	sole					
				Cluster 4				
Chicken	dove	parakeet						
				Cluster 5				
Clam	crab	crayfish	lobster	seawasp	slug	starfish	Worm	octopus
				Cluster 6				
Crow	vulture	ostrich	pheasant	wren	gull	kiwi	Flamingo	duck
Hawk	rhea	penguin	sparrow	tortoise	skimmer	lark	Swan	skua
				Cluster 7				
Flea	termite	gnat	housefly	ladybird	moth	wasp	Honeybee	
				Cluster 8				
Mole	opossum							
				Cluster 9				
Scorpion								

Another example was done with LC conceptual clustering algorithm using the database (DB) *zoo*. This DB has 101 descriptions of animals in terms of 16 attributes (15 Boolean and 1 quantitative). This DB can be consulted in http://www.ics.uci.edu/pub/machine-learning-databases/zoo.

In this example, the matching criterion for comparing all the attribute values was used. For comparing the object descriptions, we used the same similarity function of the previous example. As clustering criterion, the β_0-compact criterion with β_0=0.8 was applied. The extensional structuralization is shown in table 4. In this table, it is possible to observe that the animals in each cluster are very similar according to the set of attributes used to describe them. In cluster 1, mammals with similar characteristics appear. In cluster 2, some fishes and reptiles appear. In cluster 3, very similar fishes appear. In cluster 4, very similar domestic birds appear. In cluster 5, some crustaceans and mollusks appear. In cluster 6, non-domestic birds appear. In cluster 7, insects appear. In cluster 8, two very similar mammals appear and finally isolated in cluster 9, the scorpion appears.

Table 5. Intensional structuralization of zoo data.

Cluster	Concepts		Cluster	Concepts	
1	milk=[yes]		6	toothed=[no] legs=[2 4] domestic=[no]	(6*)
2	milk=[no] toothed=[yes] legs[0 4]		7	breathes=[yes] legs=[6]	(2*)
3	predator=[no] fins=[yes]	(2*)	8	milk=[yes] predator=[yes] catsize=[no]	(2*)
4	feathers=[yes] domestic=[yes]		9	legs=[8] tail=[yes]	(9*)
5	backbone=[no] legs=[0 4 6 5 8]	(2*)			

In table 5, you can see the concepts built by LC. The concepts give us information about how the objects in the clusters are. The clusters having more than one concept were marked with * in parenthesis and also appear the total number of concepts generated by LC. For a particular cluster, the concept informs us that there are not any objects in other clusters satisfying it (typical testor condition). In other words, the concept distinguishes the objects in a particular cluster from objects in other clusters

Table 6. Extensional and intensional structuralization for mushroom data.

Cluster	P	E	Concepts	
1	256	0	odor=[pungent]	
2	0	512	odor=[almond anise] stalk-root=[club]	
3	0	768	ring-type=[evanescent] habitat=[grasses]	(*4)
4	0	96	odor=[none] habitat=[urban]	(*2)
5	0	96	odor=[almond anise] habitat=[woods]	(*6)
6	0	192	stalk-surface-below-ring=[scaly] spore-print-color=[brown black]	(5*)
7	0	1728	gill-size=[broad] spore-print-color=[brown black] habitat=[woods]	(12*)
8	1296	0	ring-type=[large]	
9	192	0	odor=[creosote]	
10	288	0	ring-type=[pendant] spore-print-color=[chocolate]	(4*)
11	0	192	habitat=[waste]	
12	1728	0	gill-color=[buff]	
13	0	48	ring-type=[flaring]	
14	72	0	spore-print-color=[green]	
15	0	48	stalk-color-below-ring=[brown] habitat=[leaves]	(4*)
16	32	0	stalk-surface-below-ring=[scaly] population=[several]	
17	8	0	cap-color=[white] habitat=[leaves]	(2*)
18	0	192	veil-color=[brown orange]	(3*)
19	0	288	spore-print-color=[white] habitat=[grasses]	(4*)
20	0	32	stalk-color-below-ring=[white] ring-number=[two] habitat=[paths]	(12*)
21	36	0	ring-type=[none]	(5*)
22	8	0	veil-color=[yellow]	(2*)
23	0	16	stalk-color-below-ring=[brown] ring-type=[pendant]	(21*)

Finally, we present the result after apply LC to Mushroom database (see http://www.ics.uci.edu/pub/machine-learning-databases/mushroom).

Here $C(x_s(O_i), x_s(O_j)) = \begin{cases} 1 & if\ x_s(O_i) = x_s(O_j) \vee \\ & x_s(O_i) = ? \vee x_s(O_j) = ? \\ 0 & otherwise \end{cases}$ was used to manage missing values but in a particular practical problem, it must reflect the criterion of analogy employed by the expert. As clustering criterion, the β_0-conected criterion with β_0=0.95 was applied. The extensional structuralization is shown in table 6. In this case you can see that the clusters built by LC contain exclusively edible (E) or poisonous (P) mushrooms. In the table you can see the correspondent concepts to the clusters (only one for those clusters with more than one).

4. Concluding Remarks

In the paper a new conceptual clustering algorithm was presented. It can be applied in problems where the objects are described using simultaneously qualitative and quantitative attributes and there exist missing data. The output of the algorithm does not depend on the input order of objects. The clusters are formed based on properties of similarity instead of statistical or probabilistic criteria, so the concepts generated are not statistical descriptors, but logical properties based on the attributes used for describing the objects under study. In other words, our proposed algorithm generates clusters with simple conceptual interpretations.

The use of comparison criteria by attribute and its integration in a similarity function allows modeling a problem more precisely. In this way, the expert's knowledge in soft sciences can be inserted in computer systems to solve data analysis and classification problems.

Finally, the use of a genetic algorithm in order to calculate a subset of typical testors, allows obtain this subset, in a time less than a traditional algorithm, which has exponential run time complexity.

References

1. R.S. Michalski, Knowledge acquisition through conceptual clustering: A theoretical framework and an algorithm for partitioning data into conjunctive concepts. (Special issue on knowledge acquisition and induction), Policy Analysis and Information Systems 3, 1980, 219-244.
2. Fisher D, Knowledge Acquisition Via Incremental Conceptual Clustering. Readings in *Machine Learning*, Shavlik and Dietterich, eds., Morgan Kaufmann, 1990, 267-283.
3. M. Lebowitz, "Concept learning in a rich input domain: Generalization-Based Memory". *Machine Learning: An artificial Intelligence Approach, volume 2*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, eds., Morgan Kaufmann, Los Altos, CA., 1986, pp. 193-214.
4. S.J. Hanson, M. Bauer, Conceptual Clustering, Categorization, and Polymorphy. Machine Learning 3, 1989, 343-372.

5. J. Bejar, U. Cortés, LINNEO+: Herramienta para la adquisición de conocimiento y generación de reglas de clasificación en dominios poco estructurados. In: Proceedings of the III Iberoamerican Conference on Artificial Intelligence, IBERAMIA'92, La Habana, Cuba., 1992, pp. 471-482.
6. H. Ralambondrainy, A conceptual version of the k-means algorithm. Pattern Recognition Letters 16, 1995, 1147-1157.
7. J.Fco. Martínez-Trinidad, A. Guzman-Arenas, The logical combinatorial approach to pattern recognition an overview through selected works. Pattern Recognition 34/4, 2001, 1-11.
8. J.Fco. Martínez-Trinidad, J. Ruiz-Shulcloper, Fuzzy clustering of semantic spaces. Pattern Recognition 34/4, 2001, 43-53.
9. J. Ruiz-Shulcloper, J.J. Montellano-Ballesteros, A new model of fuzzy clustering algorithms. In: Proceedings of the 3th European Congress on Fuzzy and Intelligent Technologies and Soft Computing, Aachen, Germany, 1995, pp. 1484-1488.
10. J.Fco. Martínez-Trinidad, J. Ruiz-Shulcloper, M. Lazo Cortés, Structuralization of universes. Fuzzy Sets & Systems 112/3, 2000, 485-500.
11. M. Lazo-Cortés, J. Ruiz-Shulcloper, Determining the feature relevance for non clasically described objects and a new algorithm to compute typical fuzzy testors. Pattern Recognition Letters 16, 1995, 1259-1265.
12. M. Lazo-Cortés, J. Ruiz-Shulcloper, E. Alba-Cabrera, An overview of the evolution of the concept of testor. Pattern Recognition 34/4, 2001, 13-21.
13. M. Lazo-Cortés, M. Douglas de la Peña, T. Quintana-Gómez, Testors by class: An application to character recognition. Proceedings of III Iberoamerican Workshop on Pattern Recognition, March, Mexico, D.F., 1998, pp. 229-236. (In Spanish).
14. G. Sánchez, M. Lazo y O. Fuentes, 1998, "Genetic algorithm to compute typical testors with minimum weight". Proceedings of IV Iberoamerican Workshop on Pattern Recognition. Havana, Cuba. Pp. 207-213. (In Spanish).
15. G. Sánchez, 1997 "Develop and Programming of efficient algorithms (sequential and parallel) to compute typical testors on a basic matrix". MS Dissertation, BUAP, Puebla, México. (In Spanish).

Data Mining Approach Based on Information-Statistical Analysis: Application to Temporal-Spatial Data

Bon K. Sy[1] and Arjun K. Gupta[2]

[1]Queens College/CUNY, Computer Science Department, Flushing NY 11367 U.S.A.
bon@bunny.cs.qc.edu
[2]Bowling Green State University, Department of Mathematics and Statistics, Bowling Green, Ohio 43403 U.S.A.
gupta@math.bgsu.edu

Abstract. An information-statistical approach is proposed for analyzing temporal-spatial data. The basic idea is to analyze the temporal aspect of the data by first conditioning on specific spatial nature of the data. Parametric approach based on Guassian model is employed for analyzing the temporal behavior of the data. Schwarz information criterion is then applied to detect multiple mean change points --- thus the Gaussian statistical models --- to account for changes of the population mean over time. To examine the spatial characteristics of the data, successive mean change points are qualified by finite categorical values. The distribution of the finite categorical values is then used to estimate a non-parametric probability model through a non-linear SVD-based optimization approach; where the optimization criterion is Shannon expected entropy. This optimal probability model accounts for the spatial characteristics of the data and is then used to derive spatial association patterns subject to chi-square statistic hypothesis test.

1 Introduction

An example of temporal-spatial data is the monthly average temperature data. Let's assume the monthly average temperature data set of three cities in the U.S. --- Boston, Denver, and Houston --- is available. A typical task related to these temporal-spatial data analysis may attempt to answer two questions:

1. Suppose the monthly average temperature data of each city are Guassian distributed, we ask the question whether there are multiple (Guassian) mean change points of monthly average temperature data, and if so, where do they locate in the time sequence?

2. If multiple change points exist, are there any significant statistical association patterns that characterize the changes in the mean of the monthly average temperature data of the three cities?

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 128–140, 2001.

Formally, let $O_i(t_1)$ ... $O_i(t_n)$ be a sequence of n independent observations made at the ith of p possible locations; where $i=1..p$. Temporal-spatial data analysis to be discussed in this paper can be formulated as a 3-step process:

Step 1: Given a specific location indexed by i, and assuming the observations are Guassian, the specific task is to detect mean change points in the Guassian model.

Step 2: Upon detection of mean change points, the specific task is to identify the optimal non-parametric probability model --- subject to maximum Shannon entropy --- with discrete-valued random variables: $X_1, X_2, \ldots, X_p$; where each value of X_i accounts for a possible qualitative change of successive mean change points; e.g., $\{X_i: x_1 = increase, x_2 = no\text{-}change, x_3 = descrease\}$.

Step 3: Upon derivation of the optimal non-parametric probability model, the specific task is to identify statistical association pattern manifested as a p-dimensional vector of $\{v_{i,j}: i= 1.. p, j=1..3\}$; where $v_{i,j}$ represents the jth value of random variable X_i.

We will now present the problem formulation for each step. The problem formulation for step 1 is focused on the temporal aspect of the temporal-spatial data. The problem formulation for step 2 acts as a "bridge" process to shift the focus of the analysis from the temporal aspect to the spatial aspect. The problem formulation for step 3 is focused on the spatial aspect of the analysis of temporal-spatial data.

1.1 Problem Formulation 1 (for Step 1):

Let $X_1(T), X_2(T), \ldots, X_p(T)$ be p time-varying random variables. For some $i \in \{1..p\}$, let $O_i(t_1)$ ... $O_i(t_n)$ be a sequence of n independent observations of $X_i(T)$. Suppose each observation $O_i(t_j)$ is obtained from a normal distribution model with unknown mean $\mu_{i,j}$ and common variance σ, we would like to test the hypothesis:

$$H_0: \mu_{i,1} = \ldots = \mu_{i,n} = \mu_i \text{ (unknown)}$$

Versus the alternative:

$$H_1: \mu_{i,1} = \ldots = \mu_{i,c1} \neq \mu_{i,c1+1} = \ldots = \mu_{i,c2} \neq \ldots \neq \mu_{i,cq} = \ldots = \mu_{i,cq+1} = \mu_{i,n}$$

Where $1 \leq c1 < c2 < .. < cq + 1 = n$

On a fixed i, this statistical test --- if H_0 is accepted --- implies that all the observations of $X_i(T)$ belong to a single normal distribution model with mean = μ_i. In other words, $X_i(T)$ can be modeled as a Guassian model. If H_0 is rejected, it implies that each observation of $X_i(T)$ belongs to one of the q populations. In other words, $X_i(T)$ has to be modeled by q Guassian models.

1.2 Problem Formulation 2 (for Step 2):

Following the problem formulation for step 1 and assuming H_1 is not rejected, the change from $\mu_{i,j}$ to $\mu_{i,j+1}$ will be qualified as either *increase* or *decrease*; where $j \in \{c1, ..., cq\}$. For any time unit with a fixed $k' \in \{1..n\}$, we abbreviate $X_i(t_{k'})$ as X_i. Note that X_i can assume one of three discrete values according to the following rules:

$$X_i = \begin{cases} 0 & \text{if } k' \in \{c1, ., cq\} \text{ and } \mu_{i,k'} > \mu_{i,k'+1} \text{ (i..e., decrease)} \\ 1 & \text{if there is no change in the Guassian mean} \\ 2 & \text{if } k' \in \{c1, ., cq\} \text{ and } \mu_{i,k'} < \mu_{i,k'+1} \text{ (i..e., increase) .} \end{cases} \quad (1)$$

Given the marginal and joint frequency counts of the possible discrete values of $\{X_i\}$, we would like to identify an optimal discrete-valued probability model that preserves maximally the biased probability information available while minimizes the bias introduced by unknown probability information. The optimization criterion will be the Shannon expected entropy which captures the principle of minimum biased unknown information. We will later show that this problem formulation is indeed an optimization problem with linear constraints and a non-linear objective function.

1.3 Problem Formulation 3 (for Step 3):

Upon the identification of the optimal probability model, we would like to investigate the existence of statistically significant spatial patterns characterized by the joint event of $X=\{X_i : x_{ij} = j$ where $j = 0..2\}$ where $|X| = p$. Specifically, we would like to test the hypothesis:

H_0: $\{X_i : x_{ij}\}$ in X are independent of each other for $i=1..p$.

Versus the alternative

H_1: $\{X_i : x_{ij}\}$ in X are interdependent of each other for $i=1..p$.

2 Related Work

The concept of patterns is common in data mining community [1]. One notion of the concept of patterns is to capture the meaning and the quality of the information embedded in data. In this research, we attempt to apply statistical techniques for analyzing and discovering statistical patterns, and to apply information theory for interpreting the meaning behind the statistical analysis.

Measuring information content can be dated back to 1920 [2] when it was introduced by Nyquist [3]. Shannon [4] later introduced the concept of entropy based on probability measure to quantify uncertainty in disambiguating a message in communication engineering. A significant early attempt to establish a linkage between statistics and information theory is reported by Kullback [5]. The study on

specific aspects of information theory such as weight of evidence [6] and statistical interdependency [7] can be found elsewhere. The relationship between image set patterns and statistical geometry was studied by Grenander [8]. A far more extensive discussion on the general concept of patterns was published later [9], [10], [11].

Among different aspects of the concept of patterns discussed by Grenander, one interesting aspect found by the first author of this paper is the possibility of interpreting joint events of discrete random variables surviving statistical hypothesis test on interdependency as statistically significant association patterns. In doing so, significant previous works already established [6], [12], [13], [14], [15], [16] may be used to provide a unified framework for linking information theory with statistical analysis. The significance of such a linkage is that it not only provides a basis for using statistical approach for predicting hidden significant association patterns, but for using information theory as a measurement instrument to determine the quality of information obtained from statistical analysis.

3 Information-Statistical Analysis

3.1 @ Problem Formulation 1 (for Step 1):

Recall the formulation presented in section 1, for some $i \in \{1..p\}$, let $O_i(t_1) \ldots O_i(t_n)$ be a sequence of n independent observations of $X_i(T)$. Suppose each observation $O_i(t_j)$ is obtained from a normal distribution model with unknown mean $\mu_{i,j}$ and common variance σ, we would like to test the hypothesis --- for each $i \in \{1..p\}$:

$$H_0: \mu_{i,1} = \ldots = \mu_{i,n} = \mu_i \text{ (unknown)}$$

Versus the alternative:

$$H_1: \mu_{i,1} = \ldots = \mu_{i,c1} \neq \mu_{i,c1+1} = \ldots = \mu_{i,c2} \neq \ldots \neq \mu_{i,cq} = \ldots = \mu_{i,cq+1} = \mu_{i,n}$$

Where $1 \leq c1 < c2 < .. < cq + 1 = n$.

The statistical hypothesis test shown above is to compare the null hypothesis under the assumption that there is no change in the mean against the alternative hypothesis that there are q changes at the locations $c1, c2, ..., cq$. To determine whether there are multiple change points for the mean, Schwarz Information Criterion (SIC) along with binary segmentation technique [17] is employed.

Schwarz Information Criterion [18] has the form: $-2\log L(\hat{\theta}) + p \log n$, where $L(\hat{\theta})$ is the maximum likelihood function for the model, p is the number of free parameters in the model, and n is the sample size. In this setting we have one and q models corresponding to the null and the alternative hypotheses, respectively. The decision to accept H_0 or H_1 will be made based on the principle of minimum information criterion. That is, we do not reject H_0 if $SIC(n) < \min_{m<k<n-m} SIC(k)$ (where $m=1$ in this case for univariate model) and reject H_0 if $SIC(n) > SIC(k)$ for some k and estimate the position of change point k by $\hat{k}$ such that

$$SIC(\hat{k}) = \min_{m<k<n-m} SIC(k)$$

For detecting multiple change points [19], the binary segmentation technique proposed by Vostrikova can be realized as the following tasks:

Task 1:
Test for no change point versus one change point by selecting the minimum SIC among *{SIC(x)| x∈D}* where *D = {2, .., n-1}*. If $min_{x\in D\cup\{n\}} SIC(x) = SIC(n)$, then stop. There is no change point. If $min_{x\in D} SIC(x) = SIC(c1')$ where $1< c1'< n$, then there is a change point at *c1'* and we go to tasks 2. We can only find changes between *2* and *n-1*. This limitation is caused by the requirement of the existence of the maximum likelihood estimator for the problem.

Task 2:
Test the two subsequences before and after the change point at *c1'* separately for a single change.

Task 3:
Repeat the process until no further subsequences have change points.

Task 4:
The collection of change point locations found in task 1 through task 3 is denoted by *{c1', c2',..., cq'}*, and the estimated total number of change points is then *q*. Here, the estimates *c1', c2',..., cq'* are consistent for *c1, c2, ..., cq* according to Vostrikova [17].

The above steps are repeated for every value of $i \in \{1..p\}$:

Under H_0 and a given *i*,

$$SIC(n) = -2\log L(\hat{\theta}_n) + p\log n$$

With the assumption of Guassian model, *SIC(n)* becomes

$$SIC(n) = n\ log\ 2\pi + n + nlog\sigma_i + 2log\ n \qquad \textbf{(2)}$$

Where

$$\sigma_i = (1/n)\ \Sigma^n_{j=1}\ (O_i(t_j) - \mu_i)^2 \qquad \text{and} \qquad \mu_i = (1/n)\ \Sigma^n_{j=1}\ O_i(t_j)$$

Under H_1,

$$SIC(k) = -2\log L(\hat{\theta}_k) + k\log n$$

Similarly, with the assumption of Guassian model, *SIC(k)* becomes

$$SIC(k) = n \log 2\pi + n + n\log\sigma'_i + 3\log n \quad \textbf{(3)}$$

Where

$\sigma'_i = (1/n)\ \Sigma^{k}_{j=1}\ (O_i(t_j) - \mu'_i)^2 + (1/n)\ \Sigma^{n}_{j=k+1}\ (O_i(t_j) - \mu'_{n-k})^2$

$\mu'_i = (1/k)\ \Sigma^{k}_{j=1}\ O_i(t_j)$ and $\mu'_{n-k} = (1/(n-k))\ \Sigma^{n}_{j=k+1}\ O_i(t_j)$

3.2 @ Problem Formulation 2 (for Step 2):

When change point(s) is/are detected in step 1, each change point partitions the temporal-spatial data set into two sub-populations. Each population mean can be estimated following similar procedure as described in step 1. The change in the mean between two (time-wise) adjacent sub-populations can be qualified using one of three possible categorical values: *increase, same,* and *decrease*. Since each change point has a corresponding time index, not only the marginal frequency information of the corresponding "spatial-specific" random variable can be derived, the joint frequency information related to multiple variables can also be derived by alignment through common time index.

Consider the following snapshot of the categorical values of the monthly average temperature data of March from year 1970 to 1973 of three cities, Houston (HO), Denver (DE), and Boston (BO):

Table 1: Example categorical values of average temperature (month of March)

	HO	*DE*	*BO*
1970	*2*	*1*	*1*
1971	*1*	*1*	*2*
1972	*1*	*1*	*1*
1973	*1*	*2*	*0*

In the above table, "0" or "2" refers to the location of a change point and "1" refers to no change point detected. For example, two change points are detected in Boston ---- 1971 and 1973. These two change points partition the monthly average temperature data of March into three sub-populations during the period of 1970 to 1973: one sub-population is prior to 1971, one between 1971 and 1973, and one after 1973. The "2" in 1971 indicates that the Guassian mean of the model for Boston accounting the period prior to 1971 is smaller than the Guassian mean of the model for Boston accounting the period between 1971 and 1973. Similarly, the "0" in 1973 indicates that the Guassian mean of the model for Boston accounting the period between 1971 and 1973 is greater than that of the period after 1973.

With the conception just discussed, each city could be perceived as a discrete-valued random variable. The frequency count information that reflects change points indicated by "0" and "2" may be used to derive the corresponding probability distribution. For example,

$$Pr(BO:0) = \Sigma_{HO,\,DE}\, Pr(HO, DE, BO:0) = 1/4$$
$$Pr(BO:2) = \Sigma_{HO,\,DE}\, Pr(HO, DE, BO:2) = 1/4$$
$$Pr(DE:2) = \Sigma_{HO,\,BO}\, Pr(DE:2, HO, BO) = 1/4$$
$$Pr(HO:2) = \Sigma_{DE,\,BO}\, Pr(HO:2, DE, BO) = 1/4$$
$$Pr(DE:2|BO:0) = 1 \;\Leftrightarrow\; \Sigma_{HO,DE\neq 2}\, Pr(HO, DE, BO:0) = 0$$
$$\Sigma_{HO,\,DE,\,BO}\, Pr(HO, DE, BO) = 1$$

In this example, the probability model consists of $3^3 = 27$ joint probability terms $Pr(HO,DE,BO)$. There are six linear probability constraints. Given these probability constraints, we would like to derive an optimal probability model subject to

$$Max[-\Sigma_{HO,\,DE,\,BO}\, Pr(HO, DE, BO)\, log\, Pr(HO, DE, BO)].$$

The optimal solution for this example is shown below:

$Pr(HO:0, DE:0, BO:0) = 0.25$
$Pr(HO:0, DE:0, BO:1) = 0.5$
$Pr(HO:2, DE:2, BO:2) = 0.25$
$Pr(HO, DE, BO) = 0$ for the remaining joint probability terms.

The entropy of the optimal model is $Max[-\Sigma_{CH,\,DC,\,HO,\,SF,\,BO}\, Pr(HO, DE, BO)\, log\, Pr(HO, DE, BO)] = 1.5$ *bits*.

In the above example, one may wonder why we do not simply use the frequency count information of all variables to derive the desired probability model. There are several reasons due to the limitation and nature of a real world problem. Using the temperature data example, a weather station of each city is uniquely characterized by factors such as elevation of the station, operational hours and period (since inception), specific adjacent stations for data cross-validation, and calibration for precision and accuracy correction. In particular, the size of sample temperature data does not have to be identical across all weather stations. Nonetheless, the location of change points depends on each marginal individual population, and the observation on the conditional occurrence of change points of data with different spatial characteristic values (location) is still valid.

In other words, the nature of temporal-spatial data may originate from different sources. Information from different sources does not have to be consistent, and may even at times contradict each other. However, each source may provide some, but not all, information that reach general consensus, and that collectively may reveal additional information not covered by each individual.

The algorithm used for deriving the probability model just shown is based on a primal-dual formulation similar to that of the interior point method [20]. Further details are referred to a report elsewhere [21].

3.3 @ Problem Formulation 3 (for Step 3):

The purpose of deriving an optimal probability model in step 2 is to provide a basis for uncovering statistically significant spatial patterns. Our approach is to identify statistically significant patterns based on event associations. Significant event associations may be determined by statistical hypothesis testing based on mutual information measure or residual analysis.

Mutual information measure in information theory is asymptotically distributed as a chi-square distribution [5], [22]. This result has been extended elsewhere [23] to model residual analysis as a normally distributed random variable. In doing so, statistical hypothesis test based on residual analysis may be used as a conceptual tool to discover data patterns with significant event associations.

Following step 2 and using the formulation discussed earlier, let X_1 and X_2 be two random variables with $\{x_{11} \ldots x_{1z}\}$ and $\{x_{21} \ldots x_{2m}\}$ as the corresponding sets of possible values. The expected mutual information measure of X_1 and X_2 is defined as $I(X_1 X_2)= \Sigma_{i,j} Pr(x_{1i} x_{2j}) log_2 [Pr(x_{1i} x_{2j})/Pr(x_{1i})Pr(x_{2j})]$. Similarly, the expected mutual information measure of the interdependence among the multiple variables ($X_1 \ldots X_p$) is

$$I(X_1 \ldots X_p)= \Sigma^{z}_{i=1} \ldots \Sigma^{m}_{j=1} Pr(x_{1i} \ldots x_{pj}) log_2 [Pr(x_{1i} \ldots x_{pj})/Pr(x_{1i}) \ldots Pr(x_{pj})] \quad \textbf{(4)}$$

Note that the expected mutual information measure is zero if the variables are independent of each other [5], [7]. Since mutual information measure is asymptotically distributed as chi-square, statistical inference can be applied to test and compare the *null hypothesis* --- where the two variables are independent of each other --- against the *alternative hypothesis* --- where the two variables are interdependent. Specifically, the null hypothesis is rejected if $I(X_1 \ldots X_p) \geq \lambda^2/2N$; where N is the size of the data set, and λ^2 is the chi-square test statistic. The λ^2 test statistic, due to Pearson, can be expressed as below:

$$\lambda^2 = \Sigma^{z}_{i=1} \ldots \Sigma^{m}_{j=1} (o_{1i, \ldots pj} - e_{1i,..pj})^2 / e_{1i,..pj} \quad \textbf{(5)}$$

In the above equation, the λ^2 test statistic has the degree of freedom $(|X_1| - 1)(|X_2| - 1) \ldots (|X_k| - 1)$; where $|X_i|$ is the number of possible value instantiation of X_i. Here $o_{1i, \ldots pj}$ represents the observed counts of the joint event $(X_1 = x_{1i} \ldots X_p = x_{pj})$ and $e_{1i,..pj}$ represents the expected counts, and is computed from the hypothesized distribution under the assumption that $X_1, X_2, \ldots, X_p$ are independent of each other.

The chi-square test statistic and mutual information measure just shown can be further extended to measure the degree of statistical association at the event level. That is, the significance of statistical association of an event pattern E involving multiple variables can be measured using the test statistic:

$$\lambda_E^2 = (o_{1i, \ldots p,j} - e_{1i,..p,j})^2 / e_{1i,..p,j} \quad \textbf{(6)}$$

while the mutual information analysis of an event pattern is represented by $log_2[Pr(x_{1i} \ldots x_{pj})/Pr(x_{1i}) \ldots Pr(x_{pj})]$.

As suggested by Wong [23], the chi-square test statistic of an event pattern may be normally distributed. In such a case, one can perform a statistical hypothesis test to

determine whether an event pattern E bears a significant statistical association. Specifically, the hypothesis test can be formulated as below:

Null hypothesis H_0*: E is not a significant event pattern when* $\lambda_E^2 < 1.96$*,*
where 1.96 corresponds to a *5%* significance level of normal distribution.

Alternative hypothesis H_1*: E is a significant event patterns otherwise.*

In the temperature data example illustrated previously, the top three significant event patterns are *(BO:2 HO:2 DE:2)*, *(BO:1 HO:0 DE:0),* and *(BO:2 HO:0 DE:0).* However, only the pattern *(BO:2 HO:2 DE:2)* passed the chi-square λ_E^2 hypothesis test just shown.

4 Experimental Study

The information-statistical approach discussed in this paper has been applied to analyze temperature data. The temperature data source is "GHCN" data set obtained from the National Oceanic and Atmospheric Administration (NOAA) [24]. This data set consists of data collected from approximately 800 weather stations throughout the world. This data set has been repackaged. Related documents and the web accessible repackaged data can be found elsewhere [25], [26].

The temporal distribution of temperature and its variation throughout the year depend primarily on the amount of radiant energy received from the sun. The spatial distribution of temperature data depends on geographical regions in terms of latitude and longitude, as well as possible modification by locations of continents and oceans, prevailing winds, oceanic circulation, topography, and other factors. Furthermore, spatial characteristic such as elevation also plays a role on temperature changes.

In this preliminary study, ten geographical locations spanning over different regions of the United States were selected. These ten locations and the period coverage of each location are shown in the table below. The specific data set used for this study is the monthly average temperature. The size of the data available for each location varies. The longest period covers 1747 to 2000 (Boston), while the shortest period covers 1950 to 2000 (DC and Chicago).

Location	Symbol	Start year	End year	Spanning period
Chicago	CH	1950	2000	51
Washington DC	DC	1950	2000	51
Delaware	DE	1854	2000	147
Fargo	FA	1883	2000	118
Houston	HO	1948	2000	53
Kentucky	KT	1949	2000	52

Boston	BO	1747	2000	254
San Francisco	SF	1853	2000	148
St. Louis	SL	1893	2000	108
Seattle	SE	1947	2000	54

In each one of the ten locations, the change point detection analyses are carried out twelve times, one for each month using all available data. For example, the size of Jan monthly average temperature data of Boston is 254 (2000 – 1747 +1). All 254 Jan monthly average temperature data are used for detecting the change points (indexed by year) in Jan. This is then repeated for every month from Feb to Dec; where a new set of 254 data points are used for change point detection. This is then repeated for each one of the ten locations. A complete summary of all change points detected according to Schwarz information criterion using formula 2 and 3 will be reported in an extended version of forthcoming paper.

After the change points are identified, we ask the question whether any change points from various locations align in time. In other words, if there is a mean change in one location, are there any other locations also experience a mean change at the same time (by year) on a specific month? And particularly, are any change points common to at least three different locations?

Using previous formulation $O_i(t_1)$ to represent the monthly average temperature of location i at the year t_1, there are three possibilities. At the year t_1, it could be no change point, or a change point with increased Guassian mean, or a change point with decreased Guassian mean in a location i. Since there are ten locations, the number of possible combinations to account for the existence and type (increase/decrease) of change points is $3^{10} = 59049$. Obviously the problem will be unmanageable if we attempt to derive a probability model to account for the occurrences of all joint change points. Instead, we decided to study the temperature change points in 5 groups of 5 locations. They are:

Group 1:
CH (Chicago) |DC (Washington DC)|DE (Delaware)|FA (Fargo) |BO (Boston)

Group 2:
CH (Chicago) |BO (Boston) |HO (Houston) |KT (Kentucky) |DC

Group 3:
DE (Delaware)|SF (San Francisco) |FA (Fargo) |KT (Kentucky) |SE (Seattle)

Group 4:
CH (Chicago) |DE (Delaware) |FA (Fargo) |KT (Kentucky) |SL (St. Louis)

Group 5:
HO (Houston) |SF (San Francisco) |KT (Kentucky)|SL (St. Louis) |DC

In studying each of the five groups, we are interested in any trend patterns of simultaneous change points of at least three locations. With these patterns, we proceed to the following three tasks:

1. Based on the frequency count information, estimate the conditional probability of simultaneous change points.

2. Based on the conditional probability information, derive an optimal probability model with respect to Shannon entropy.
3. Based on the optimal probability model, identify statistical significant association patterns that characterize the type of changes (increased/decreased) in the Guassian mean based on Chi-square test statistic discussed in section 3 (equation 6).

5 Discussion

A question related to the issue of global warming is whether the analysis accomplished so far provides any evidence from the association patterns about the trend of global warming.

As temperature is generally cyclical by year, we would expect in an ideal case --- without global warming and without "man-made" environmental disturbance --- that there is no (Guassian) mean change, by month, in the monthly average temperature over years. However, change points are identified in every one of the twelve analyses specific to a month.

With the change points, we would like to know whether there is global warming driving the temperature raising. If so, we expect to observe an upward trend in the mean temperature indicated by the numerous occurrences of "2" comparing to "0" as defined in equation 1. This is not shown in the 12 analyses over the entire period of years being examined. Nonetheless, there seems to be localized temporal trend patterns. For example, the analysis using the data of the month of "May" shows a distinct temperature downward trend during the period 1935 – 1955. The analysis using the data of the month of "Nov" shows a distinct temperature upward trend during the period 1953 – 1963, and two distinct spikes of temperature increase between 1966 and 1985. Furthermore, the analysis using the data of the month of "March" shows a dense fluctuation in the mean temperature in comparison to other months.

If we now shift our focus to the spatial characteristics of the data, we can ask a similar question about the existence of any localized spatial trend patterns on the temperature data. By examining the significant event association patterns that also appear as the three most probable joint events in each probability model of the five studies, each study reveals some interesting observations.

In the first study group, we find that the mean temperature decrease occurred in both Chicago and DC do not just happen independently according to statistical interdependency test, and so as the mean temperature increase in both Delaware and Boston. The consistent pair-wise mean temperature change in Delaware and Boston is consistent with our expectation since they are in a relatively close geographical proximity. In the second study group, a similar phenomenon about the decrease in the mean temperature is also observed in both Boston and Chicago. An interesting contrast is the fifth study group. It shows the change in mean temperature moves in opposite direction between two locations --- San Francisco and St. Louis.

The third and fourth study groups perhaps are most interesting. In the third study group the association patterns including Delaware and Kentucky reveal a decrease in

the mean temperature while in the fourth study group the association patterns including Delaware and Kentucky reveal an increase in the mean temperature. A further study shows that both locations are in the jet stream routes --- a unique meteorological phenomenon in the United States, and both are in the close proximity of isotherm --- line of equal temperature.

6 Conclusion

This paper discussed a treatment on the temporal-spatial data based on information-statistical analysis. The analysis consists of three steps. Under the assumption of Guassian and *iid*, the temporal aspect of the data is examined by determining the possible mean change points of the Guassian model through a statistical hypothesis test using Schwarz information criterion. Based on the detected change points, we qualified the magnitude changes in the mean change points and marginalized such frequency information over the temporal domain. After doing so, the analytical step involves formulating an optimization problem based on available frequency information in an attempt to derive an optimal discrete-valued probability model that captures possible spatial association characteristics of the data. Chi-square hypothesis test is then applied to detect any statistically significant event association patterns. Preliminary result on applying the proposed method to analyze global temperature data is also reported.

Acknowledgement

Preparation of the manuscript and web-based data hosting resources are supported in part by a NSF DUE CCLI grant #0088778.

References

1. Fayyad, U. M. and Piatetsky-Shapiro, G. and Smyth, P.: From Data Mining to Knowledge Discovery: An Overview, in Advances in Knowledge Discovery and Data Mining, (editors: Fayyad, U. M. and Piatetsky-Shapiro, G. and Smyth, P. and Uthurusamy, R.). AAAI Press / MIT Press, (1996) Chapter 1, 1-34.
2. Jumarie, G., Relative Information: Theory and Applications, Springer-Verlag (1990).
3. Nyquist H., Certain Topics in Telegraph Transmission Theory, A.I.E.E. Transaction, V. 47, April (1928).
4. Shannon, C.E. and Weaver, W., The Mathematical Theory of Communication, University of Urbana Press, Urbana (1972).
5. Kullback S., Information and Statistics, Wiley and Sons, NY (1959).

6. Good I.J., Weight of Evidence, "Correlation, Explanatory Power, Information, and the Utility of Experiments," Journal of Royal Statistics Society, Ser. B, 22, (1960) 319-331.
7. Goodman L.A., "The analysis of cross-classified data: Independence, quasi-independence and interactions in contingency tables with and without missing entries," Journal of the American Statistical Association, 63:1091-1131 (1968).
8. Grenander U., Pattern Analysis: Lectures in Pattern Theory: V2, Applied Mathematical Sciences 24, Springer-Verlag, ISBN 0-387-90310-0 (1978).
9. Grenander U., Chow Y. Keenan K.M., *HANDS*: A Pattern Theoretic Study of Biological Shapes, Springer-Verlag, New York (1991).
10. Grenander U., General Pattern Theory, Oxford University Press (1993).
11. Grenander U., Elements of Pattern Theory, The Johns Hopkins University Press, ISBN 0-8018-5187-4 (1996).
12. Kullback S. and Leibler R., "On Information and Sufficiency," Ann. Math. Statistics, 22 (1951) 79-86.
13. Kullback S., Information and Statistics, Wiley and Sons, NY, (1959).
14. Haberman S.J. "The Analysis of Residuals in Cross-classified Tables," Biometrics, 29 (1973) 205-220.
15. Cover T.M. and Thomas J.A., Elements of Information Theory, Wiley (1991).
16. Chen J. and Gupta A.K., "Information Criterion and Change Point Problem for Regular Models," Technical Report No. 98-05, Department of Mathematics and Statistics., Bowling Green State U., Ohio (1998).
17. Vostrikova L. Ju., "Detecting disorder in multidimensional random process," Soviet Math. Dokl., 24, 55-59, (1981).
18. Schwarz G., "Estimating the dimension of a model," Ann. Statist., 6, 461-464 (1978).
19. Gupta, A.K. and Chen, J., "Detecting Changes of mean in Multidimensional Normal Sequences with Applications to Literature and Geology," Computational Statistics, 11:211-221, 1996, Physica-Verlag, Heidelberg (1996).
20. Wright S., Primal-Dual Interior-Point Methods, SIAM, ISBN 0-89871-382-X (1997).
21. Sy B.K., "Probability Model Selection Using Information-Theoretic Optimization Criterion," Journal of Statistical Computing and Simulation, Gordon and Breach Publishing Group, NJ, 69(3), (2001).
22. Fisher R.A., "The conditions under which χ^2 measures the discrepancy between observation and hypothesis," Journal of the Royal Statistical Society, 87 (1924) 442-450.
23. Wong A.K.C. and Wang Y., "High Order Pattern Discovery from Discrete-valued Data," IEEE Trans. On Knowledge and Data Engineering, 9(6), (1997) 877-893.
24. www http://www.ncdc.noaa.gov/wdcamet.html
25. www http://bonnet3.cs.qc.edu/weather/
26. www http://bonnet2.cs.qc.edu:1099/data_warehouse/plsql/rd_start.main

A Hybrid Tool for Data Mining in Picture Archiving System

Petra Perner[1] and Tatjana Belikova[2]

[1] Institute of Computer Vision and Applied Computer Sciences IBaI, Arno-Nitzsche-Str. 45, 04277 Leipzig, Germany
ibaiperner@aol.com http://www.ibai-research.de
[2] Institute for Information Transmission Problems Russian Academy of Sciences, B. Karetniy 19, 101447 Moscow, Russia
belikovat@mail.ru and/or belik@iitp.ru

Abstract. A tool and a methodology for data mining in picture archiving systems are presented. It is intended to discover the relevant knowledge for picture analysis and diagnosis from the database of image descriptions. Knowledge engineering methods are used to obtain a list of attributes for symbolic image descriptions. An expert describes images according to this list, and stores descriptions in the database. Digital image processing can be applied to improve imaging of specific image features or to get expert-independent feature evaluation. Decision tree induction is used to learn the expert knowledge, presented in the form of image descriptions in the database. Constructed decision tree presents effective models of decision-making, which can be learned to support image classification by the expert. A tool for data mining and image processing is presented. The developed tool and methodology have been tested in the task of early differential diagnosis of pulmonary nodules in lung tomograms and was effective for preclinical diagnosis of peripheral lung cancer, so that we applied the developed methodology of data mining in other medical tasks such as lymph node diagnosis in MRI and investigation of breast MRI.

1 Introduction

Radiology departments are in the center of fundamental changes in technologies for image acquisition and handling. The radiographic films, which have been used for analysis and diagnosis since 1895, are being replaced now by digital images, acquired in new imaging modalities, such as CT, MRI, etc. A concept of Picture Archiving and Communication Systems (PACS) have been proposed at early 80th to provide efficient and cost-effective analysis, exchanging, storing, and retrieving diagnostic images [1], [2]. PACS incorporates several subsystems and use different technologies for acquisition, storage, transmission, processing and displaying medical images, presented in digital forms. The main objective of the system is to supplying the user with easy, fast, reliable access to images and associated diagnostic information [2], [3], [4]. During the past 10 years, the technologies related to the entire PACS components became mature, and their applications have gone beyond radiology to the

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 141□156, 2001.

entire health care delivery system, so that PACS technologies gained widespread acceptance both in special clinical applications and in large-scale hospital-wide PACS.

PACS gives the user means to surpass current diagnostic ability thanks to the achievements of Computer-Assisted Radiology (CAR), such as multi-modality imaging and multimedia displaying of medical data, image processing and computer-assisted diagnoses [5], [6], [7]. CAR approach to computer-assisted diagnosis is based on image processing and pattern recognition methods. The image is treated as a two-dimensional signal, and values of some formalized features (statistics, color, characters of object shape, size etc.) are measured directly in the image and used for object classification. The main problem is to choose the features, which could properly describe medical objects. Such methods fail when we have to create a classifier on the base of expert knowledge and non-formal subjective estimations of features by the expert.

Pictures, stored in PACS archive, provide new possibilities for deep studying of specific and temporal features of the lesion and for dynamical studying of the feature evolution. That s why further development of CAD is associated with the use of new intelligent capabilities, such as data mining, which allow discovering the relevant knowledge for picture analysis and diagnosis from the database of image descriptions [8], [9], [10], [11], [12], [13], [14]. The application of data mining will help to get some additional knowledge about specific features of different classes and the way in which they are expressed in the image. This method can elicit non-formalized expert knowledge; automatically create effective models for decision-making, and can help to find some inherent non-evident links between classes and their imaging in the picture. It can help to get some nontrivial conclusions and predictions on the base of image analysis. The new knowledge obtained as a result of data analysis in the database can enhance the professional knowledge of the expert. This knowledge can be also used for teaching novices or can support image analysis and diagnosis by the expert.

Additional advantage of data mining application for decision of medical tasks is a long-run opportunity for creation of fully automatic image diagnosis systems that could be very important and useful in the case of the lack of knowledge for decision-making.

In this paper, we present our methodology for performing data mining in picture archiving systems. In Section 2, we describe the recent state of the art in image mining and the problems concerned with image mining. A design of image-mining tools is considered in Section 3. The developed tool for image mining is presented in Section 4. A methodology for data mining that has been created and tested in the task of early differential diagnosis of pulmonary nodules is describe in Section 5. Finally, in Section 6, we summarize our experience in applications of data mining methodology in different medical task, such as preclinical diagnosis of peripheral lung cancer on the basis of lung tomograms, lymph node diagnosis and investigation of breast diseases in MRI. Conclusions and plans for future work are given in Chapter7.

2 Background

As shows the analysis of the literature, the most of the recent works on image mining are devoted to knowledge discovery. They are dealing with a problem of searching the regions of special visual attention or interesting pattern in a large set of image, e.g. in CT and MRI image sets [15], [16] or in satellite images [17]. Usually experienced experts have discovered this information. However, the amount of images, which is being created by modern sensors, makes necessary the development of methods that can decide this task for the expert. Therefore, standard primitive features that are able to describe the visual changes in the image background are being extracted from the images and the significance of these features is being tested by sound statistical test [15], [17]. Clustering is applied in order to explore the images to seek for the similar groups of spatial connected components [18] or for similar groups of objects [16].

The measurement of image features in these regions or patterns gives the basis for pattern recognition and image classification. Computer vision researches are fulfilled to create proper models of objects and scene, to obtain image features and to develop decision rules that allow one to analyze and interpret the observed images. CAD methods of image processing, segmentation, and feature measurements are fruitfully used for this purpose [5], [6], [7]. The mining process is done bottom-up. As much numerical features as possible are extracted from the images in order to achieve the final goal - the classification of the objects. However, such a numerical approach usually does not allow the user to understand the way in which the reasoning process has been done.

The second approach to pattern recognition and image classification is an approach based on symbolical description of images made by the expert. This approach can present to the expert in the explicit form the way in which the image has been interpreted. The experts having the domain knowledge usually prefer the second approach.

Usually simple numerical features are not able to give description of complex objects and scenes. They can be described by an expert with the help of non-formalized symbolical descriptions, which reflect some gestalt in the expert domain knowledge. A problem is how to find out the relevant descriptions of the object (or the scene) for its interpretation, and how to construct a proper procedure for extraction of these features. This top-down approach is the more practical approach for most medical applications. However symbolical description of images and feature estimation face with numerous difficulties:

1. A skilled expert knows how to interpret the image, but often he has no well-defined vocabulary to describe the objects, visual patterns and gestalt variances, which are standing behind his diagnostic decisions. When the expert is asked to make this knowledge explicit he/she usually cannot specify and verbalize it.
2. Although numerous efforts are going on to develop such a vocabulary for specific medical tasks (for example, ACR-BIRADS-code has been constructed for image analysis in mammography) the problem of difference between "displaying and naming" still exists.

3. A developed description language will differ from a medical school to a medical school, as a result the obtained symbolical description of image features by a human will be expert-depended and subjective.
4. Besides this, the developed vocabulary usually consists of a large number of different symbolical features (image attributes) and features values. It is not clear a-priori if all the attributes, included into the vocabulary, are necessary for the diagnostic reasoning process. To select the necessary and relevant features would make the reasoning process more effective.

We propose a methodology of data mining that allows one to learn a compact vocabulary for the description of medical objects and to understand how this vocabulary is used for diagnostic reasoning. This methodology can be used for a wide range of image diagnostic tasks.

Developed methodology takes into account the recent status of the art in image analysis and combines it with new methods of data mining. It allows us to extract quantitative information from the image when it is possible, to combine it with subjectively determined diagnostic features, and then to mine this information for the relevant diagnostic knowledge acquisition by objective methods such as data mining.

Our methodology should help to solve some cognitive, theoretical and practical problems:

1. It will reproduce and display a decision model of an expert for specific tasks solution.
2. It will show the pathway of human reasoning and classification. Image features, which are basic for correct decision by the expert, will be discovered.
3. Developed model will be used as a tool to support decision-making of physician, who is not an expert in a specific field of knowledge. It can be used for teaching novices.

The application of data mining will help to get some additional knowledge about specific features of different classes and the way in which they are expressed in the image. It could help to find some inherent non-evident links between classes and their imaging in the picture that could be used to make some nontrivial conclusions and predictions on the base of elicited knowledge.

3 Design Considerations

We developed a tool for data mining, which could meet several requests:

1. The tool has to be applicable for a wide range of image diagnostic tasks and image modalities that occur in the radiological practice.
2. It should allow the medical staff to develop their own symbolic descriptions of images in the terms, which are appropriate to the specific diagnostic task.
3. Users could have a possibility for updating or adding features according to new images or a diagnostic problem.

4. It should support the user at the analysis and interpretation of images; for example at the evaluation of new imaging devises and radiographic materials.

Taking into account these criteria and the recent state-of-the-art in image analysis we provided an opportunity for semiautomatic image processing and analysis to enhance imaging of diagnostically important details on the image and to measure some image features directly in the image and by this way to supports the user by the analysis of images. The user has to have possibilities to interact with the system to do adaptation to the results of image processing.

This image-processing unit should provide extraction of such low-level features as blobs, regions, ribbons, lines, and edges. On the basis of these low-level features, we are able to calculate then some high-level features to describe the image. Besides that, the image-processing unit should allow evaluation of some statistical image properties, which might give valuable information for the image description.

However, some diagnostically important features, for example, such as "irregular structure inside the nodule", "tumor" are not so called low-level features. They present some gestalts of expert domain knowledge. Development of an algorithm for extraction of such image features can be a complex, or sometimes unsolvable problem. So, we identify different ways of representing the contents of an image that belongs to different abstraction levels. We can describe an image:

- by statistical properties that is the lowest abstraction level;
- by low-level features and their statistical properties such as regions, blobs, ribbons, edges and lines. It is the next higher abstraction level;
- by high-level or symbolic features that can be obtained from the low-level features;
- and, finally, by expert symbolic description, which is the highest abstraction level.

The image-processing unit combined with the data evaluation unit should allow a user to learn the relevant diagnostic features and effective models for the image interpretation. Therefore, the system as a whole should meet the following criteria:

1. Support the medical person by the extraction of the necessary image details as much as possible.
2. Fulfill measurement of the feature values directly in the image, when it is possible.
3. Display the interesting image details to the expert.
4. Store in a database the measured feature values as well as the subjective description of images by the expert.
5. Import these data from the database into the data-mining unit.

4 System Description

Fig.1 shows a scheme of a Picture Archiving System combined with the developed tool for data mining.

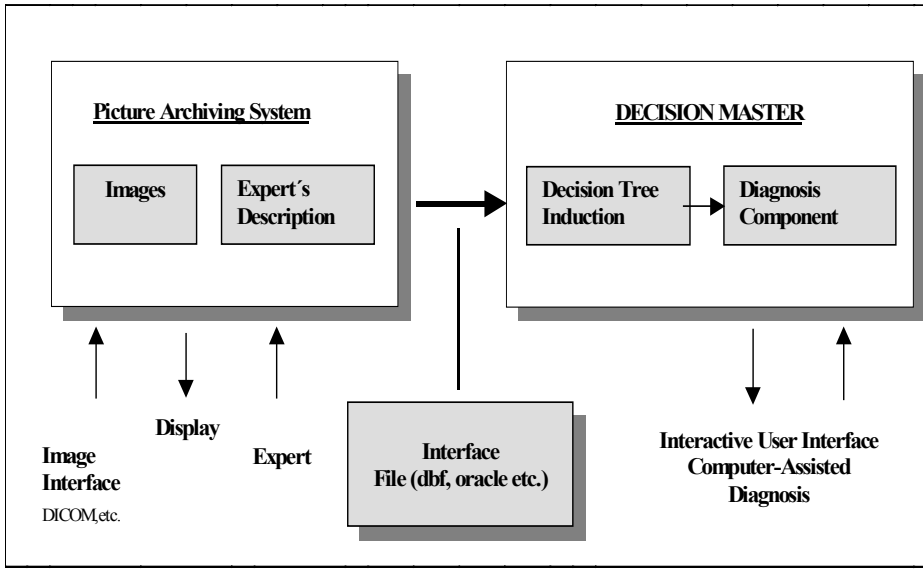


Fig. 1. A scheme of a Picture Archiving System combined with the data-mining tool

There are two parts in the tool: the unit for image analysis (Fig.2) and the unit for data mining (Fig.3).

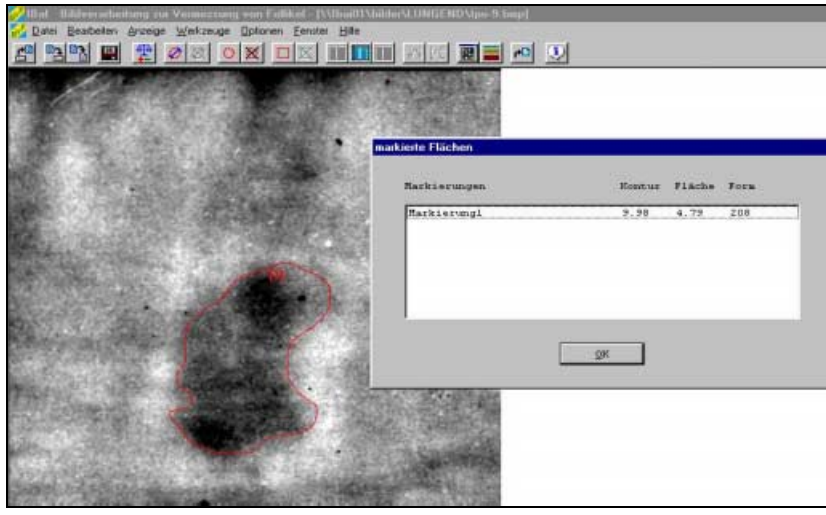


Fig. 2. Interactive Image Analysis Tool

Both units are written in C++ and runs under Windows95 and Windows NT. These two units communicate over a database of image descriptions, which is created in the

frame of image processing unit. This database is the basis for the data-mining unit (Fig.3).

An image from the image archive is selected by the expert and then it is displayed on a monitor (Fig. 2). To perform image processing an expert communicates with a computer. He/she determines whether the whole image or its part have to be processed and outlines an area of interest (or a nodule region) with an overlay line. The parameters of optimal filter are then calculated automatically. Afterwards the expert can calculate some image features in the marked region (object contour, square, diameter, shape, and some texture features) [19]. The expert evaluates or calculates image features and stores their values in a database of image features. Each entry in the database presents features of the object of interest. These features can be numerical (calculated on the image) and symbolical (determined by the expert as a result of image reading by the expert). In the latter case, the expert evaluates object features according to the attribute list, which has to be specified in advance for object description. Then he/she inputs these values into the database.

When the expert has evaluated a sufficient number of images, the resulting database can be used for the mining process. The stored database can be easily loaded into the data mining tool *Decision Master* (Fig. 3).

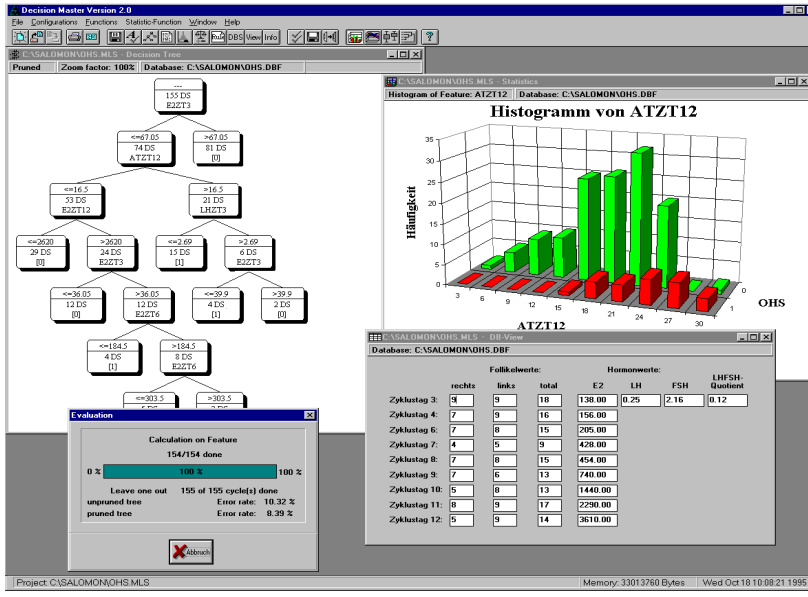


Fig. 3. Data Mining Tool Decision Master

Decision Master fulfills a decision tree induction that allows one to learn a set of rules and basic features necessary for decision-making in a specified diagnostic task. The induction process does not only act as a knowledge discovery process, it also works as a feature selector, discovering a subset of features that is the most relevant to the problem solution.

Decision trees partition decision space recursively into sub-regions based on the sample set. By this way the decision trees recursively breaks down the complexity of the decision space. The outcome has a format, which naturally presents the cognitive strategy that can be used for human decision-making process.

For any tree, all paths lead to a terminal node corresponding to a decision rule that is a conjunction (AND) of various tests. If there are multiple paths for a given class, then the paths represent disjunctions (ORs) [20].

The developed tool allows choosing different kinds of method for feature selection, feature discretization, pruning of the decision tree and evaluation of the error rate. It provides an entropy-based measure, gini-index, gain-ratio and chi square method for feature selection [21].

Decision Master provides the following methods for feature discretization: cut-point strategy, chi-merge discretization, minimum description length principal based discretization method and lvq-based method [21]. These methods allow one to make discretization of the feature values into two and more intervals during the process of decision tree building. Depending on the chosen method for attribute discretization, the result will be a binary or n-ary tree, which will lead to more accurate and compact trees.

Decision Master allows one to chose between cost-complexity pruning, error reduction based methods and pruning by confidence interval prediction. The tool also provides functions for outlier detections.

To evaluate the obtained error rate one can choose test-and-train and n-fold cross validation. Missed values can be handled by different strategies [21].

The user selects the preferred method for each step of the decision tree induction process. After that, the induction experiment can start on the acquired database. A resulting decision tree will be displayed to the user. He/she can evaluate the tree by checking the features used in each node of the tree and comparing them with his/her domain knowledge.

Once the diagnosis knowledge has been learnt, the rules are provided whether in txt-format for further use in an expert system or the expert can use the diagnosis component of *Decision Master* for interactive work. It has a user-friendly interface and is set up in such a way that non-computer specialists can handle it very easily.

5 Status Report

Image processing methods [22], [23] and data mining methods [24], [25] have been used to perform image analysis, feature description and data mining in the task of early differential diagnosis of pulmonary nodules on the basis of linear lung tomograms.

Two physicians supported our experiment. One was a high skilled pulmonologist, who had a long practice in analysis and classification of processed images. The other one also was a pulmonologist, but he had no special courses of processed image reading and interpretation.

For our experiment, we used a database of lung tomograms of 175 patients with verified diagnosis. Patients with small pulmonary nodules (up to 3 cm) have been

selected (Set1: 64 cases of benign disease and 111 cases of peripheral lung cancer). Conventional (linear) coronal plane tomograms with 1 mm thickness of section were used for specific diagnosis of solitary lung nodules. For our test experiment we selected 38 images (Set2: 20 malignant and 18 benign nodules). About a half of these images were referred to as complex cases as they yielded ambiguous diagnostic decisions during the analysis of the unprocessed images by the experts.

Original linear tomograms were digitized with step of 100 micron (5 line pairs per millimeter) to get 1024 x 1024 x 8 bits matrices with 256 levels of gray.

The use of linear tomograms and such a digitization enabled an acquisition of high spatial resolution of anatomical details that were necessary for the specific diagnosis.

5.1 Image Processing

To improve results of specific diagnosis of small solitary pulmonary nodules we used optimal digital filtering [22], [23] and the analysis of the post-processed images.

An optimal filter has been developed to improve imaging of the objects of interest. We designed the filter on the basis of expert's domain knowledge: we discussed with the expert what parts of image (what objects, details, structures) were diagnostically important for the diagnosis of peripheral lung cancer. The remainder image (lung tissues) was regarded as a "background" in this medical task. Filtering has to emphasize diagnostically important details in the medical image so that the physician could be more certain in reading and interpretation of image features.

No formal model of "the useful signal" was available in this task. So the only possible way was to model a background. We developed an optimal linear filter (Wiener filter), which eliminated the background and by this way improved imaging of informative part of image. Several background modes [22] have been selected and tested. One of the model has been selected, which satisfied several expert's criteria:

1. from the radiologist's point of view it gave the best imaging of important details;
2. it didn't input artifacts, which could reduce diagnostic accuracy;
3. all details in the processed images were in concordance with morphological observations.

X-ray-morphological comparisons [23] have been fulfilled to confirm that the developed filter satisfied these criteria.

5.2 Attribute List and Image Description

Image processing enhanced imaging of diagnostically important features, which were then described by the expert and stored in the database of image descriptions.

First, an attribute list was set up together with the expert. The list covered all possible attributes, used for diagnosis by the expert, as well as the corresponding attribute values, see Table 1. We learned our lesson from another experiment [24] and created an attribute list having no more than three attribute values. Otherwise, the resulting decision tree is hard to interpret and the tree building process stops very soon because of the splitting of the data set into subsets according to the number of attribute values.

Table 1. Attribute List

Attribute	Short Name	Attribute Values
Class	*CLASS*	1 malignant 2 benign
Structure inside the nodule	*STRINSNOD*	1 Homogeneous 2 Inhomogeneous
Regularity of Structure inside the nodule	*REGSTRINS*	1 Irregular Structures 2 Regular orderly
Cavitation	*CAVITATIO*	0 None 1 Cavities
Areas with calcifications inside the nodule	*ARWCAL*	0 None 1 Areas with calcifications
Scar-like changes inside the nodule	*SCARINSNOD*	0 None 1 Possibly exists 2 Irregular fragmentary dense shadow
Shape	*SHAPE*	1 Nonround 2 Round 3 Oval
Sharpness of margins	*SHARPMAR*	1 NonSharp 2 MixedSharp 3 Sharp
Smoothness of margins	*SMOMAR*	1 NonSmooth 2 MixedSmooth 3 Smooth
Lobularity of margins	*LOBMAR*	0 NonLobular 1 Lobular
Angularity of margins	*ANGMAR*	0 Nonangular 1 Angular
Convergence of vessels	*CONVVESS*	1 Vessels constantly 2 Vessels are forced away the nodule 3 None
Vascular Outgoing Shadows	*VASCSHAD*	0 None 1 Chiefly vascular
Outgoing sharp thin tape-lines	*OUTSHTHIN*	0 None 1 Outgoing sharp thin tape-lines
Invasion into surrounding tissues	*INVSOURTIS*	0 None 1 Invasion into surrounding tissues
Character of the lung pleura	*CHARLUNG*	0 No Pleura 1 Pleura is visible
Thickening of lung pleura	*THLUNGPL*	0 None 1 Thickening
Withdrawing of lung pleura	*WITHLUPL*	0 None 1 Withdrawing
Size of Nodule	*SIOFNOD*	Numbers (e. g, 1.2) in cm

A radiologist watches the processed image (see Fig.2) displayed on-line on a TV monitor, evaluates its specific features (character of boundary, shape of the nodule, specific objects, details and structures inside and outside the nodule, etc.), interprets these features according to the list of attributes, and inputs the codes of appropriate attribute values into the database answering to the computer requests.

Hard copies of the previously processed images from the archive were used in this work as well. The collected data set was passed as a dBase-file to the inductive machine learning tool.

5.3 Decision Tree Induction

Decision tree induction was then used to learn the expert knowledge, presented in the form of image descriptions. Constructed decision tree provided discovering of basic features and creation of decision-making models, which could be learned to support image classification by the expert.

We used the developed tool *Decision Master* [25], which realized decision tree induction method, and created binary-trees based on information gain criteria [26]. Pruning is done based on reduced-error pruning technique [27]. Evaluation was done by 10-fold cross-validation.

The unpruned tree consists of 20 leaves, see Fig.4, the pruned tree consists of 6 leaves, see Fig.5.

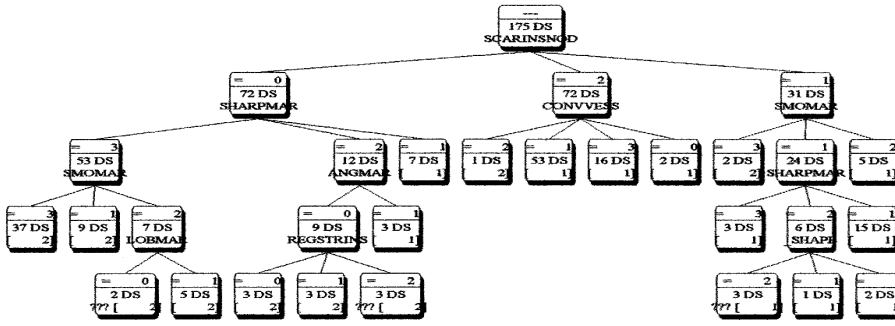


Fig. 4. Decision Tree (unpruned) for Set 1

Our expert liked the unpruned tree much more since nearly all attributes he is using for decision-making appeared in the tree. The expert told us that the attribute *Structure* is very important, also the *attribute Scar-like changes inside the nodule*.

However the expert wonders why other features such as *Structure* and some others didn't work for classification. The expert told us that he usually analyzes a nodule starting with its *Structure*, then tests *Scar-like changes inside the nodule*, then *Shape* and *Margin*, then *Convergence of Vessels* and *Outgoing Shadow in Surrounding tissues*.

Although decision trees represent the decision in a comprehensible format to human, the decision tree might not represent the strategy used by an expert since it is

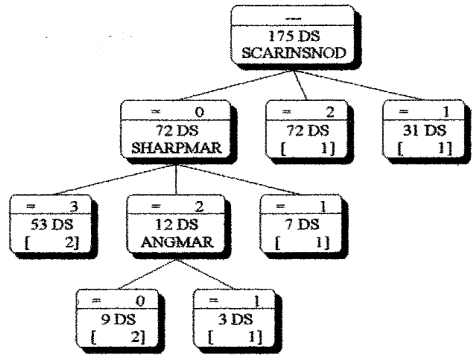


Fig. 5. Decision Tree (pruned) for Set 1

always the attribute appearing first in database and satisfying the splitting criteria that is chosen.

Therefore, we investigated the error rate as main criterion (see Table 2, Table 3). Table 2 shows the error rate for the learnt decision model calculated with cross-validation (1) and the error rate calculated with test-and-train (2).

Table 2. Error Rate: (1) For Cross Validation; (2) Evaluation of Decision tree on Test Data

Error Rate before pruning.		Error Rate after pruning
(1)	6,857 %	
(2)	6,30%	7,428 % 7,3 %

Table 3. Comparisons between Human Expert and Decision Tree Classification

Accuracy		*Sensitivity/Specificity*			
		Class_1		Class_2	
Human	DT	Human	DT	Human	DT
94,4 %	93,2 %	97,5 %	96,2 %	91,4 %	90 %

Besides the error rate we calculate Sensitivity for Class1 and Specificity for Class2, which are error criteria usually required for medical applications: $E_{Sens} = S_{C1m}/N_{C1}$, $E_{Spec} = S_{C2m}/N_{C2}$, where S_{C1m} is the number of mis-classified samples of Class1 and N_{C1} the number of all samples of Class 1, and S_{C2m} and N_{C2} are respectively the same for Class2.

These experiments showed that the learnt classifier comes close to the performance of the human expert.

Table 4. Comparison between Human Expert and Decision Tree Classification: (1) High-level Expert; (2) Middle-level Expert

Accuracy		*Sensitivity/Specificity*			
		Class_1		Class_2	
Human	DT	Human	DT	Human	DT
94,5%	95,7%	96,2%	93,65%	90%	99%
55,2%	73%	61,1%	74%	50%	72,5%

In Table 4 there are results of the high-level expert, non-trained (middle level) expert, and the decision tree classifier. As the middle level expert did not know how to read a new roentgenological picture that appeared after digital image processing, it brought much uncertainty and noise into the data. The resulting error rate shows that classifier based on decision tree gives reliable error rate even in the case of bad (noisy and incomplete data) obtained as a result of image readings, see Table 4.

6 Lessons Learned

We have found out that our methodology of data mining allows a user to learn the decision model and the relevant diagnostic features. A physician can independently use such a methodology of data mining in practice. He/she can easily perform different experiments until he/she is satisfied with the result. By doing that he/she can explore his application and find out the connection between different knowledge pieces.

However some problems should be taken into account for the future system design.

As we have already pointed out in Chapter 5 an expert tends to specify symbolical attributes with a large number of attribute values. For e.g. in a previous experiment [24] the expert specified for the attribute "margin" fifteen attribute values such as "non-sharp", "sharp", "non-smooth", "smooth", and so on. A large number of attribute values will result in small sub-sample sets soon after the tree building process started. It will results in a fast termination of the tree building process. This is also true for small sample sets that are usual for medicine. Therefore, a careful analysis of the attribute list should be done after the physician has specified it.

During the process of building the tree, the algorithm picks the attribute with the best attribute selection criteria. If two attributes have both the same value, the one that appears first in the attribute list will be chosen. That might not always be the attribute the expert would choose himself. To avoid this problem we think that in this case we should allow the expert to choose manually the attribute that he/she prefers. We expect that this procedure will bring the resulting decision model closer to the expert ones.

The described method of data mining had been already established in practice. It runs at the University hospital in Leipzig and Halle and at the Veterinary department of the University in Halle, where the method is used for analysis of sheep follicle,

evaluation of imaging effect of radiopaque material for lymph nodule analysis, mining knowledge for IVF therapy, transplantation medicine and for the diagnosis of breast carcinoma in MR images. In all these tasks we did not have a well-trained expert. These were new tasks and reliable decision knowledge has not been built up in practice yet.

The physicians were very happy with the obtained results, since the learnt rules gave them deeper understanding of their problems and helped to predict new cases. It helped the physicians to explore their data and inspired them to think about new improved ways of diagnosis.

7 Conclusion and Further Work

In this paper we presented our methodology of data mining in picture archiving systems. The basis for our study is a sufficiently large database with images and expert descriptions. Such databases result from the broad use of picture archiving systems in medical domains.

We were able to learn the important attributes needed for image interpretation and to understand the way in which these attributes were used for decision-making by applying data mining methods to the database of image descriptions. We showed how the domain vocabulary should be set up in order to get good results, and which techniques should be used in order to check reliability of the chosen features.

The explanation capability of the induced tree was reasonable. The attributes included into the tree represented the expert knowledge.

Finally, we can say that picture archiving systems in a combination with data mining methods open a possibility of advanced computer-assisted medical diagnosis system development. However, it will not give the expected result if the PACS have not been set up in the right way. Pictures and experts descriptions have to be stored in a standard format in the system for further analysis. Since standard vocabulary and very good experts are available for many medical diagnosis tasks this should be possible. If the vocabulary is not a priori available, then the vocabulary can be determined by a methodology based on the repertory grid [28]. What is left is to introduce this method to the medical community that we have done recently for mammogram analysis and lymph nodule diagnosis. Unfortunately, it is not possible to provide image analysis systems, which can extract features for all kind of images. Often it is the case that it is not clear how to describe a particular feature by automatic procedures developed for image feature extraction. The expert s description will still be necessary for a long time. However, once the basic discriminating features have been found the result can lead in the long run to fully automatic image diagnosis system, which is set up for specific type of image diagnosis. In our future work we like to extend the number of feature extractors to a larger number of necessary feature extractors.

Acknowledgements

We like to thank the medical experts Prof. Heywang-Kbrunner from the university of Halle; Dr. N. Yashunskaya and Dr. O. Ogrinsky from Moscow Medical Academy and veterinary expert Dr. Kaulfu from the university of Halle for the expert analysis and description of medical images, used in our experiments with lung tomograms, breast MR images and sheep follicle images.

References

1. In: Proceedings of 15th Symposium for Computer Applications in Radiology: Filmless radiology reengineering the practice of Radiology for the 21st Century. Baltimore, USA 1998. J. of Digital Imaging, Vol. 11, 3, Suppl. 1 (1998).
2. Andriole, K. P.: Anatomy of Picture archiving and communication systems: Nuts and Bolts-Image Acquisition: getting digital images from imaging modalities. J. of Digital Imaging, Vol. 12, 2, Suppl. 1 (2000) 216-217.
3. In: Proceedings of 16th Symposium for Computer Applications in Radiology. PACS: Performance Improvement in Radiology. Houston, USA 1999. J. of Digital Imaging, Vol. 12, 2, Suppl. 1 (2000).
4. In: Proceedings of 17th Symposium for Computer Applications in Radiology: The electronic Practice: Radiology and Enterprise. Philadelphia, USA 2000. J. of Digital Imaging, Vol 13, 2, Suppl. 1 (2000).
5. In: Proceedings of SPIE International Symposium Medical Imaging 1998, San-Diego, USA. SPIE, Vol. 3338 (1998).
6. Proceedings of SPIE International Symposium Medical Imaging 2000, San-Diego, USA. SPIE , Vol. 3981. (2000).
7. In: Proceedings of 14th Int. Congress on Assisted Radiology and Surgery CARS 2000, San-Francisco, USA. Int. Congress Series, Vol. 1214. Excerpta Medica (2000).
8. Heywang-Kbrunner, S., Perner, P.: Optimized Computer-Assistet Diagnosis based on Data Mining, Expert Knowledge and Histological Verification. IBaI Report ISSN 1431-2360 (1998).
9. Perner, P. A.: Knowledge-based image inspection system for automatic defect recognition, classification, and process diagnosis. Int. J. on Machine Vision and Applications 7 (1994) 135-147.
10. Boose, J. H., Shema, D. B., Bradshaw, J.M.: Recent progress in Aquinas: a knowledge acquisition workbench. Knowledge Acquisition 1 (1989) 185-214.
11. Kehoe, A. and Parker, G.A.: An IKB defect classification system for automated industrial radiographic inspection. IEEE Expert Systems 8 (1991) 149-157.
12. Schrder, S., Niemann, H., Sagerer, G.: Knowledge acquisition for a knowledge based image analysis system. In: Proc. of the European Knowledge-Acquisition Workshop (EKAW 88). Bosse, J., Gaines, B. (eds.), GMD-Studien, Vol. 143, Sankt Augustin (1988).
13. Kolodner, J .L., Simpson, R. L., Sycara, K.: A Process Model of Case Based Reasoning in Problem Solving. In: Proc. 9th Int. Joint conf. on Artificial Intelligence. Los Angeles, CA, (1985) 100-110.
14. Perner, P.: Case-Based Reasoning for the Low-level and High-level Unit of an Image Interpretation System. In: Advances in Patter Recognition. Singh S. (ed.). Springer-Verlag (1998) 45-54.

15. Megalooikonomou, K., Davatzikos, C., Herskovits, E.: Mining lesion-defect associations in a brain image database, in Proc. Int. Conf. Knowledge Discovery and Data Mining (KDD-99), San Diego, California, August 1999, 347-351, 1999.
16. Eklund, P. W., You, J., Deer, P.: Mining remote sensing image data: an integration of fuzzy set theory and image understanding techniques for environmental change detection. In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology. Belur V. Dasarathy (eds.). SPIE , Vol. 4057 (2000) 265-273.
17. Burl, M. C., Lucchetti, D.: Autonomous visual discovery. In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology, Belur V. Dasarathy (eds.). SPIE, Vol. 4057 (2000) 240-250.
18. Zaiane, O. R., Han, J.: Discovery spatial associations in Image. In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology. Belur V. Dasarathy (eds.), SPIE, Vol. 4057 (2000) 138-148.
19. Zamperoni, P.: Feature Extraction, In:, Progress in Picture Processing, Eds. H. Maitre and J. Zinn-Justin. Elsevier Science (1996) 121-184.
20. Weiss, S.: Predictive Data Mining, Kluwer Verlag (1996).
21. Perner, P.: Data Mininig on Multimedia Data, Springer Verlag (2001) (to appear).
22. Belikova, T. P., Yashunskaya, N. I,. Koganm, E.A. Computer-Aided differential Diagnosis of Small solitary Pulmonary Nodules. Computer and Biomedical Research, Vol. 29**,** 1 (1996) 48-62.
23. Belikova, T .P., Yashunskaya, N. I,. Kogan, E. A.: Computer analysis for differential diagnosis of small pulmonary nodules. In: Proc. of Int. Congress for lung cancer. Athens Greece, Monduzzi. Editore. Intern. (1994) 93-98.
24. Perner, P., Belikova, T. P., Yashunskaya, N. I. Knowledge Acquisition by symbolic decision tree induction for interpretation of digital images in radiology. In: Advances in Structural and Syntactical Pattern Recognition. Lecture Notes in Computer Science. Perner, P, Wang, P., Rosenfeld, A. (eds). Springer, Vol. 1121 (1996). 208-219.
25. Data Mining Tool Decision Master. http://ww.ibai_solution.de .
26. Baird, H. S., Mallows C. L.: Bounded-Error in Pre-classification Trees. In: Shape, Structure and Pattern Recognition. Dori ,D., Bruckstein, A (eds). World Scientific Publishing Co, (1995) 100-110.
27. Quinlain, J. R.: Simplifying decision tree. Intern. Journal on Man-Machine Studies, Vol. 27, (1987) 221-234.
28. Perner, P.: How to use Repertory Grid for Knowledge Acquisition in Image Interpretation. HTWK Report 2 (1994).

Feature Selection for a Real-World Learning Task

D. Kollmar and D.H. Hellmann

Universität Kaiserslautern, D 67663 Kaiserslautern, Germany
{kollmar, hellmann}@rhrk.uni-kl.de

Abstract. Machine learning algorithms used in early fault detection for centrifugal pumps make it possible to better exploit the information content of measured signals, making machine monitoring more economical and application-oriented. The total amount of sensors is reduced by exhausting the information derived from the sensors far beyond the scope of traditional engineering through the application of various features and high-dimensional decision-making. The feature selection plays a crucial role in modelling an early fault detection system. Due to presence of noisy features with outliers and correlations between features a correctly determined subset of features will distinctly improve the classification rate. In addition the requirements for the hardware to monitor the pump decrease therefore its price. Wrappers and filters, the two major approaches for feature selection described in literature [4] will be investigated and compared using real-world data.

1 The Machine Learning Task

The process industry ensures many commodities of our modern society like chemicals, pharmaceutics and nutrition. In most plants pumps are the driving force and their availability directly determines the production output. Smaller process pumps are low price, highly standardised products; hundreds and even thousands of them are employed in one large plant alone. Due to the diversity of the processes and the media handled, the actual design of each pump as well as its operating range vary depending on the type of plant. Mostly, pump damage is not only caused by ageing and natural wear, but also by impermissible operating conditions such as operation at minimum flow or dry-running. Market competition has put a focus on the decrease of operating costs, namely production outages due to pump downtimes and plant design costs which are strongly driven by redundant pumps and the additional piping and instrumentation required therefore.

Traditional machine monitoring techniques employ one or more sensors per fault whose signals are interpreted by a human expert. Sterile, hazardous or toxic environments put strong impositions on sensors; the price of some measurement chains almost equals the pump's price. In former times skilled service staff detected faults such as a bearing failure or cavitation from alterations of the sound emitted by a pump even with complex mixtures caused by adjacent machinery.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 157–172, 2001.

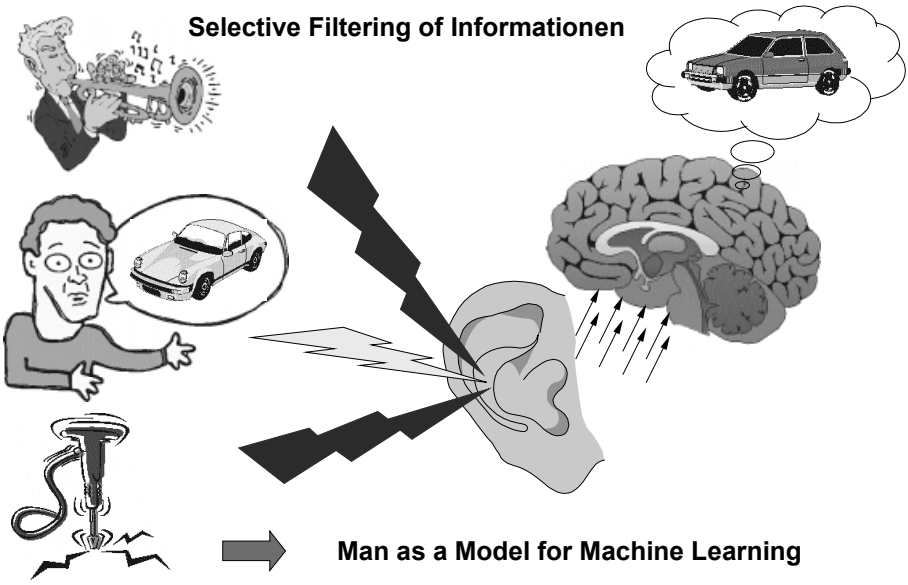


Fig. 1. Extracting Information from Complex Signals

This implicit monitoring system fulfils the requirements of sensor minimisation and even does not require any media-wetted sensors which could conflict with certain aggressive products. Therefore the human expert is the ideal model for the early fault detection system.

2 Machine Learning Meets Pump Monitoring

In the DFG-funded project 'Machine Learning applied to early fault detection for failure critical components' [12] the application of machine learning algorithms for pump monitoring has been investigated for the first time. The interdisciplinary collaboration of pump experts and computer scientists was a major asset in this new approach. The data is obtained from velocity probes mounted on the pump casing. The amplitude and phase frequency spectra carry a huge amount of information despite strong correlation's between certain frequencies.

The machine learning algorithm proposed for this project is See5 [11] a very fast and powerful inducer for decision trees. Many comparative studies of machine learning algorithms as by Lim and Loh [5] or the Statlog Project [7] proved the competitive classifying ability and the outstanding computing performance of this program. A recent study by Martens et. al. [6] comparing C4.5, the precessor of See5, and NEFClass, a late neural network with fuzzy logic, reflects these results:

Neither of the algorithms has a constantly superior classification rate; depending on the type of learning problem the neural network or the decision tree is better. On average the decision tree even outperforms the neural network regarding the classification rate. As shown by Quinlan in [9] neural networks are advantageous,

when most features are to be exploited in parallel for making up a decision, whereas the decision tree can handle features, which are only locally relevant. The major advantage of decision trees is their training time which is 2 – 3 orders of magnitude shorter compared to neural networks.

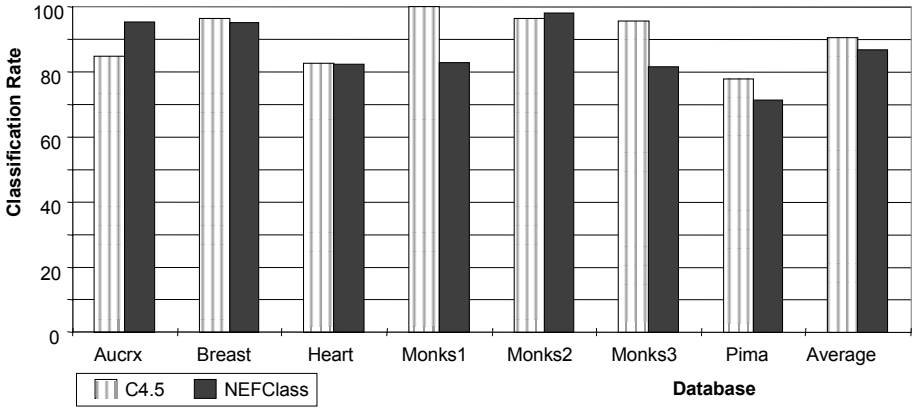


Fig. 2. Comparison of C4.5 and NEFClass on various machine learning problems [6]

A further asset of See5 are cost matrices which allow the user to model a real-world problem The individual importance of the correct classification of different classes as well as the ratio of false alarms and undetected faults can be influenced. Centrifugal pumps cease to operate when the medium contains too much gas. The gas intensity of the liquid is a continuous variable and any threshold for defining classes is arbitrary.

predicted class

true class	B	GA	GB	GC	K1	K2	N
B	0	3	3	3	3	3	3
GA	3	0	1	2	3	3	1
GB	3	1	0	1	3	3	3
GC	3	2	1	0	3	3	3
K1	3	3	3	3	0	2	1
K2	3	3	3	3	2	0	3
N	3	2	3	3	2	3	0

Fig. 3. Cost matrix for modeling a pump monitoring task in See5

As shown in figure 3 costs can be employed to model the physical relationship among the different gas classes (GA, GB and GC). 0 refers to a correct classification, 1 describes a misclassification into a neighbouring class and 3 refers to a total misclassification. Blockage (pressure sided valve closed) is a purely binary fault which is translated into misclassification costs of 3 for any different class. During optimisation of a pump monitoring system, single cost values can be adjusted to

influence the classification ability of the decision tree. Although the overall error rate often increases the errors can be shifted to harmless misclassifications, e. g. the confusion of neighbouring classes.

In the following study the cost matrix remains unchanged in order to ensure the comparability of the results. The average costs which in the following will be used to evaluate different classifiers are obtained by dividing the cumulated costs for the unseen test data sets (3 cross validations) by their population.

The process of modelling an early fault detection system is described in figure 4. The pump is equipped with different sensors and operated under normal operating conditions at various flow rates and under every fault to be detected. In a preliminary study only the frequencies carrying information are preselected by a human expert in order to reduce the amount of data. Up to 40 features are generated from one sensor. A typical database consists of 30.000 datasets with 30 features each.

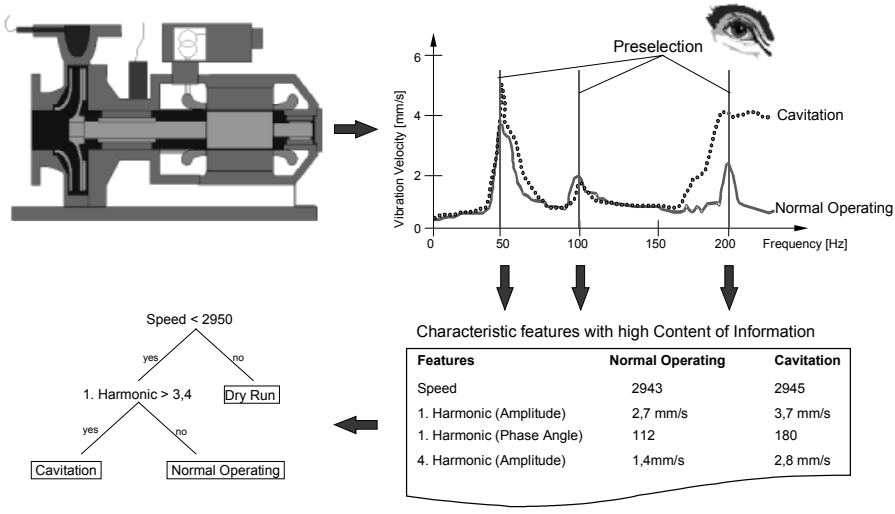


Fig. 4. Defining features from spectra and using them for growing a decision tree

3 Wrappers for Feature Selection

As shown in figure 5, an exhaustive exploration of the feature space is hardly possible with as few as 10 features due to the number of combinations arising and the resulting computing time.

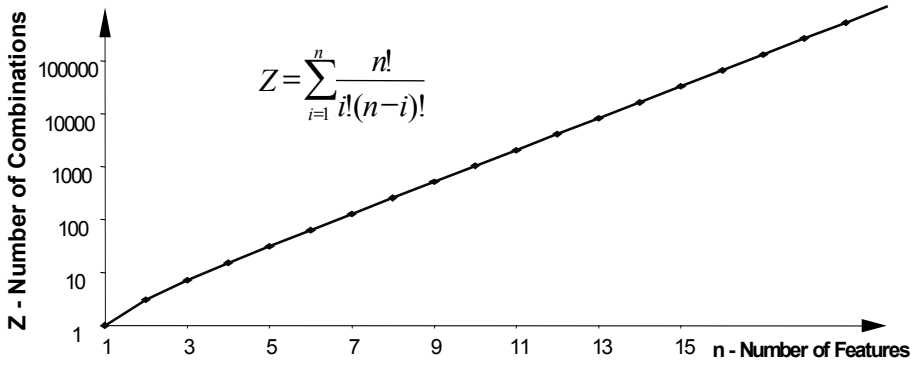


Fig. 5. Evolution of the number of different features subsets for an exhaustive search

Wrappers [4, 8] are auxiliary algorithms wrapped around the basic machine learning algorithm to be employed for classification. They create several different subsets of features and compute a classifier for each of them. According to a criterion, mainly the error rate on unseen data, the optimal subset is composed.

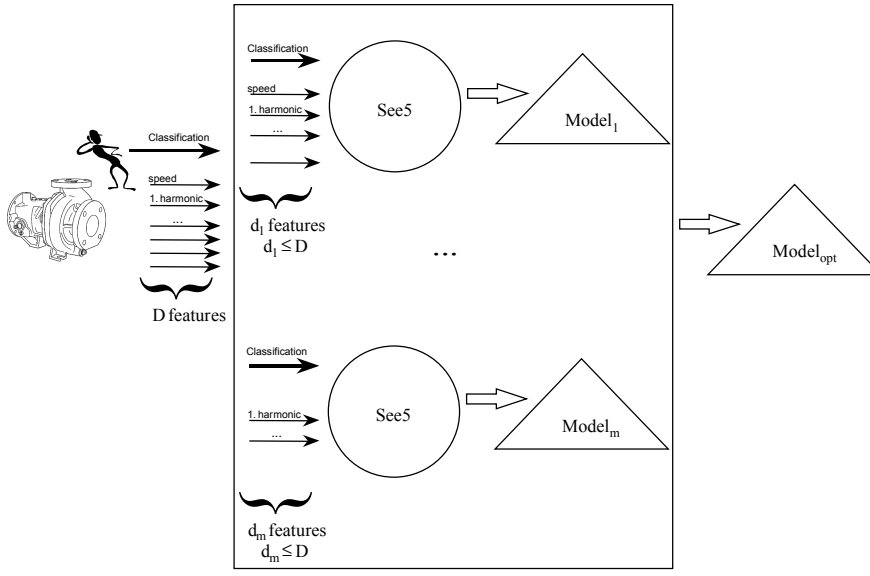


Fig. 6. Wrappers determine the optimal feature subset by running the machine learning algorithm with a sequence of subsets

3.1 Best Features (BF) [2]

Given a data set with N features, N classifiers are trained, each using one feature only. The classifiers are ranked according to the criterion and the best M < N are elected. Best Features does not take into account dependencies between features (feature A is

only effective with presence of feature B) nor redundancy among the features and is therefore a fast but very sub optimal approach.

3.2 Sequential Forward Selection (SFS) [2]

In its basic form the feature selection starts with an empty subset. In each round all possible feature subsets are constructed, which contain the best feature combination found in the round before plus one extra feature. A classifier is trained and evaluated based on each subset. This approach is capable of avoiding the selection of redundant features.

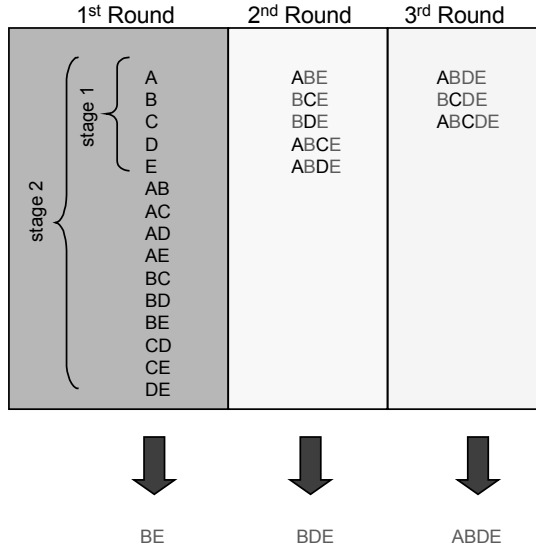


Fig. 7. Double-stage sequential forward-algorithm with 5 features (A, B, C, D, E)

In order to determine dependencies between features, several features have to be introduced simultaneously, the basic scheme is depicted in figure 7. With 25 features the first round requires training of 25 classifiers in a single-stage SFS, 325 classifiers in a double-stage SFS and as many as 2625 classifiers in a triple-stage SFS.

3.3 Sequential Backward Selection (SBS) [2]

The SBS is initialised with the complete feature set. In each round all possible feature subsets are constructed from the current subset omitting one feature. After each round the feature subset with the best criterion is retained. As in the case of the SFS, the single-stage version is capable of detecting redundant features but higher stage orders have to be considered for dealing with dependent features.

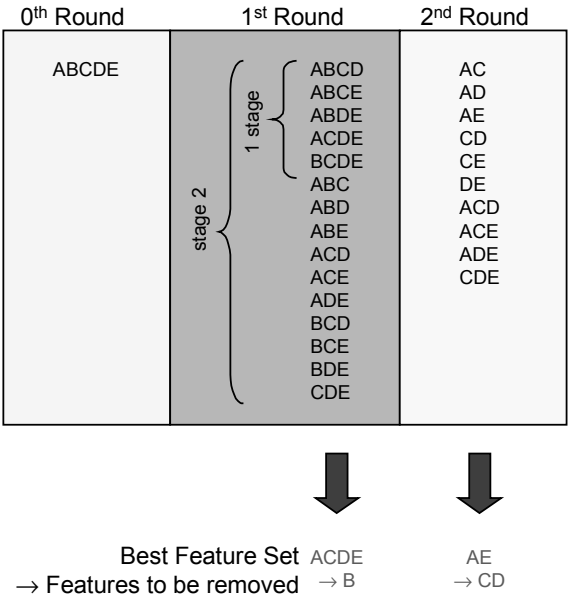


Fig. 8. 2-stage sequential Backward-Algorithm with 5 features (A, B, C, D, E)

3.4 Branch and Bound (BB) [2]

In contrary to the selection algorithms described so far, the BB yields an optimal subset (as by an exhaustive search) provided the following assumption is fulfilled: Adding an extra feature y to a feature subset ξ does not lead to a deterioration of the criterion J(ξ) (monotonicity)

Feature Sets:

$$\{y_1, y_2, ..., y_D\} = Y = \underline{\xi}_0 \supset \underline{\xi}_1 \supset \underline{\xi}_2 \supset ... \supset \underline{\xi}_D = \{\ \} \tag{1}$$

Criterion:

$$J\left(\underline{\xi}_0\right) \geq J\left(\underline{\xi}_1\right) \geq J\left(\underline{\xi}_2\right) \geq ... \geq J\left(\underline{\xi}_D\right) \tag{2}$$

The BB-algorithm is depicted in figure 9.

The tree is covered from the right to the left side. The criterion c = J({1,2} of the first leaf {1,2} is determined. Next the tree is searched for the first unprocessed node. Departing from this node the rightmost, unprocessed path is chosen. For each node the criterion is determined. If the criterion J({1,3,5,6} < c = J {1,2}, the current path is abandoned.

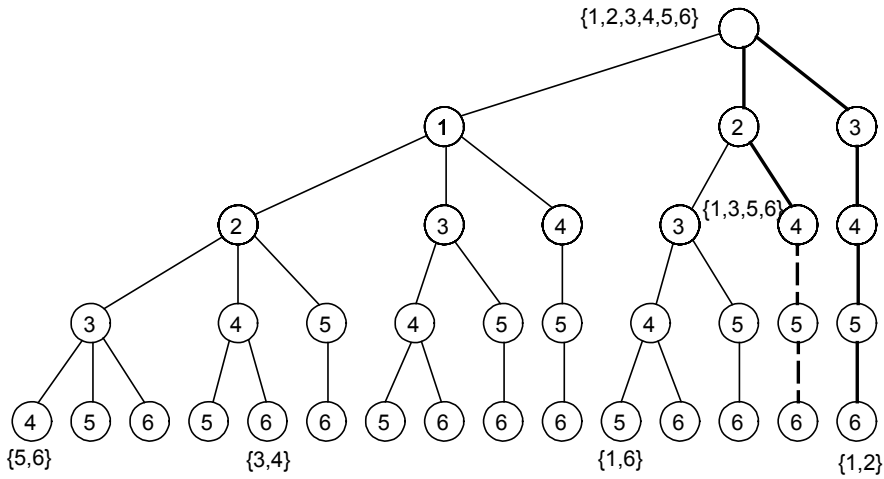


Fig. 9. Branch and Bound-Tree for determining the optimal subset containing 2 out of 6 features

BB is ideal for any classificator which allows for a recursive training when adding or removing a feature, e. g. linear discriminant analysis. In general the monotonicity-assumption only holds when the error rate on training data is chosen as criterion but fails on any criterion based on unseen test data. In this case BB is not optimal any more.

4 Comparison of Wrappers

As BF and BB were considered unsuitable for the pump data, only SBS and SFS have been examined. The results depicted in figure 10 show a strong influence of feature selection on the quality of the resulting classifier. Whereas using all 25 features (i. e. directly employing the data without feature selection) yields average misclassification costs of 0,62, the optimum for a single stage SFS occurs at 17 features with costs of 0,50, but a combination of only 6 features leads to costs of 0,51. A double-stage SFS further reduces the misclassification costs to 0,38. Both SBS trials proved worse than their SFS counterparts although SBS theoretically should retain dependent features due to its deselection strategy.

A ranking of the features (figure 11) showed the varying effectiveness of the features depending on the selection method. The feature c1 was selected in the first round of the double -stage SFS but in the 21st round of the single-stage SFS. What is more the second feature selected aside c1 in the double -stage selection was n, which was selected in the 2nd round of the double-stage SFS. The chart shows a strong interdependency between the features and the non-linear behaviour of the inducer.

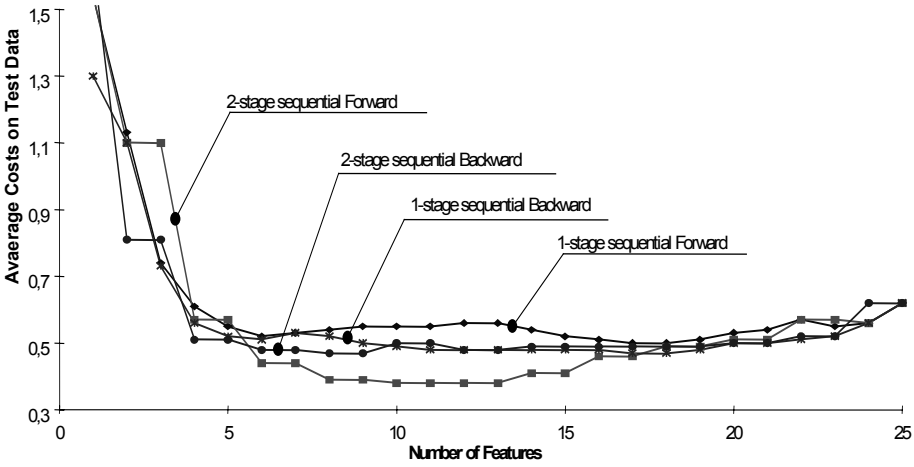


Fig. 10. Comparison of SBS and SFS with pump data (30.000 data sets; 25 features; 3 fold crossvalidation averaged in each round) using See5 with a cost matrix (weighted misclassification rate)

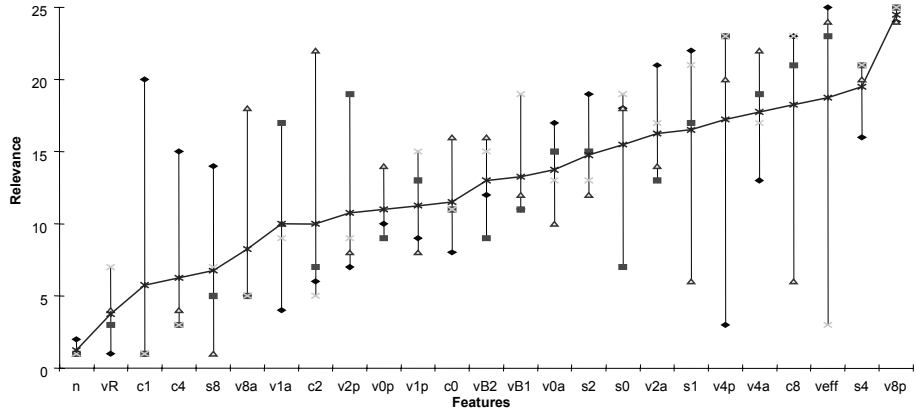


Fig. 11. Ranking of the 25 features in the 4 selections shown in figure 10 (1 = most relevant)

A further obstacle is the influence of the two tuning parameters implemented in See5 on the actual feature selection. Not only the value of the criterion is changed but also the location of the optimum and the best feature subset are altered.

5 Genetic Wrapper

The comparison of wrappers showed a distinct advantage of multistage wrappers due to their ability of selecting dependent features. Even if the selection is stopped after reaching the optimum, the computing effort remains high because of the high

amount of combinations in the beginning. A further disadvantage of the sequential selection is the rigid strategy which does not allow for the removal of features which, in the course of the selection process, have become obsolete.

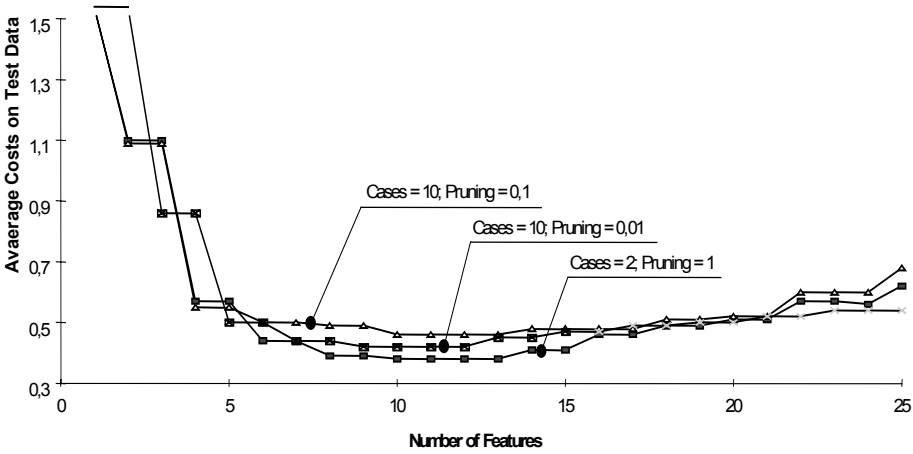


Fig. 12. Influence of the parameters in See5 on a double -stage SFS (30.000 data sets; 25 features; 3 fold crossvalidation averaged in each round). The curve for Pruning=0,01 was started with a triple-stage SFS in the 1st round and then continued as a double -stage SFS

An alternative approach consists of a genetic selection which explores the feature space from a determined or from a random starting point.

Genetic algorithms use two parallel strategies for optimising:

- evolution (small and steady development of the population and selection of the best)
- mutation (spontaneous creation of new specimen)

The flow charts of the genetic selection are shown in figures 13 & 14. The algorithm uses a result stack for the subsets already evaluated and a working stack for new subsets. The subsets in the working stack are sorted with respect to their prognosis, which is the evaluation criterion reached by their parent. In each round the first A subsets are evaluated.

The evolution is modelled by single-stage SFS and SBS. After training a classifier from a subset with M features, all possible sequential combinations with M+1 and M-1 features are composed. All subsets which have not been evaluated yet are stored in the working stack.

The algorithm will fully exploit the feature space, if no stopping criterion is introduced. If memory is scarce, the size of the working stack can be limited by deleting the subsets having the worst prognosis. When the selection is attracted by a local minimum, the area will be fully explored before the search is continued outside unless a mutation leads to a better subset.

The double stage SFS, applied to the pump data, has determined a subset of 10 features with an average cost on test data of 0,38. The least computing effort (stop immediately after a minimum) would have been the training of 1386 decision trees.

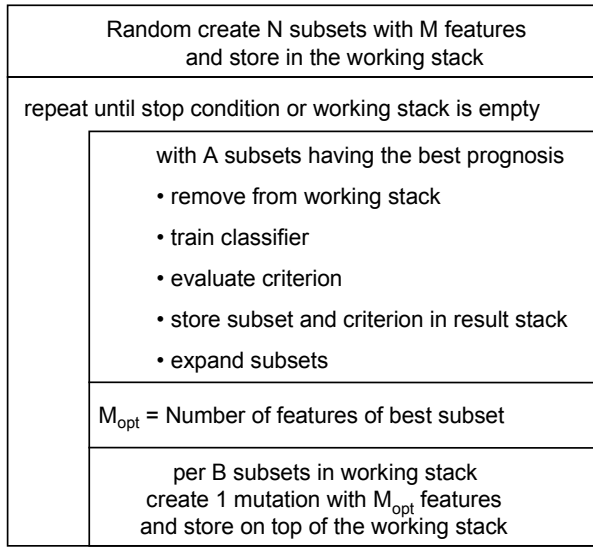


Fig. 13. Flow chart for genetic feature selection

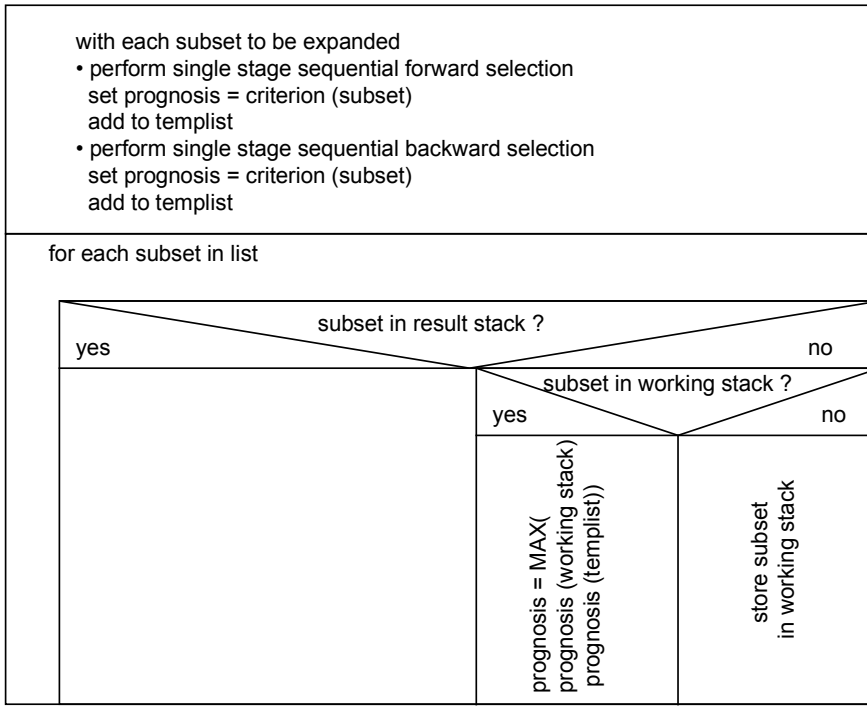


Fig. 14. Expansion of Subsets

After training 610 decision trees the genetic selection (figure 15) comes up with a subset of 9 features at a cost of 0,38. Within 1110 subsets evaluated (Round 12) there are 5 subsets with 9 or 10 features having costs of 0,38 each. The user can choose between different subsets and therefore minimise other criteria imposed on the features such as number or costs of sensors required.

Due to the evolutionary strategy the random selected features play a minor role for convergence and quality of the results. The computing time rises if the number of features in the initial selection is far off the optimum, as shown in figure 16. For an unknown dataset it may be worth to start with a single stage SFS and set the initial value M to the number of features at minimum criterion.

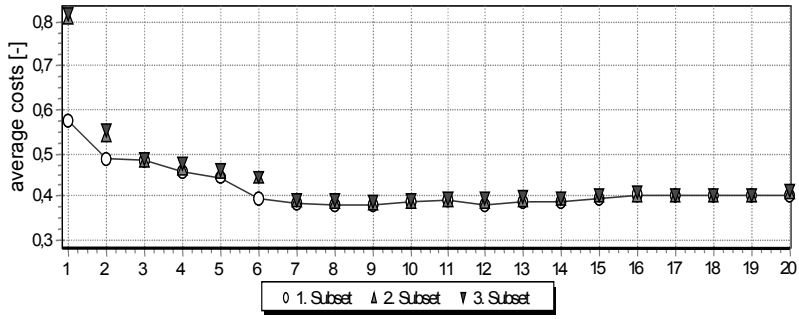


Fig. 15. Genetic feature selection with N = 10 subsets / M = 10 features / A = 100 subsets / B = 2%

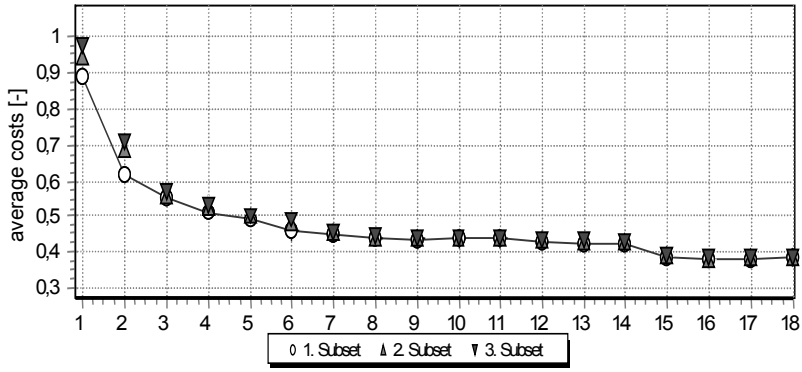


Fig. 16. Genetic feature selection with N = 10 subsets / M = 4 features / A = 100 subsets / B = 2%

6 Filters for Feature Selection

Filters are standalone algorithms which determine an optimal subset regardless of the machine learning algorithm envisaged for classification. Filters are mainly used for ML algorithms with categorical classes. The effectiveness of continuous features for continuous classification problems can be directly evaluated using regression techniques. Three groups of filters can be discerned:

- Wrappers combined with a fast machine learning algorithm, e. g. Nearest Neighbour. The evaluation of the subsets is based on a criterion defined for the classifier of the secondary ML algorithm.
- Blackbox-algorithms, which directly determine a suitable feature subset (e. g. Eubafes [10]).
- Evaluation of the feature subset based on the class topology in the feature space. Criteria are the distance between the classes in the subspace (based on the records in the training set) or the classification probability based on the estimated density functions of the classes, e. g. Chernoff, Bhattacharyya or Mahalanobis distance [2].

Criteria which depend on distances in the feature space are strongly influenced by scaling (e. g. due to an altered amplification of a signal used to derive some features) and transformations applied to features as well as the chosen metric δ:

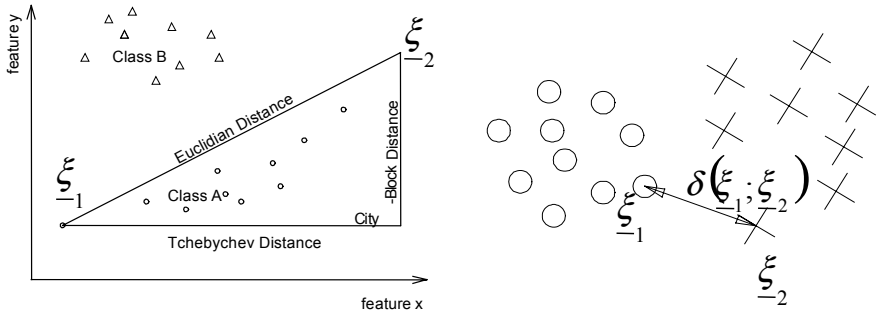


Fig. 17. Metrics for the distance between 2 points ζ_1 and ζ_2 in the feature space

In general the Euclidean metric is preferred, but City-Block (sum of components) Tchebycheff (largest component) deliver different kinds of information about the data at modest computing effort.

The estimation of the probabilistic distance requires the knowledge of the class density function. In general a normal distribution is assumed but particularly in the case of multi-modal class densities this assumption is ill-conceived.

6.1 Analysis of Decision Trees

Occasionally the usage of statistics based on decision trees is used as a filter. In [1] the features are ranked according to their frequency of occurrence in the decision tree.

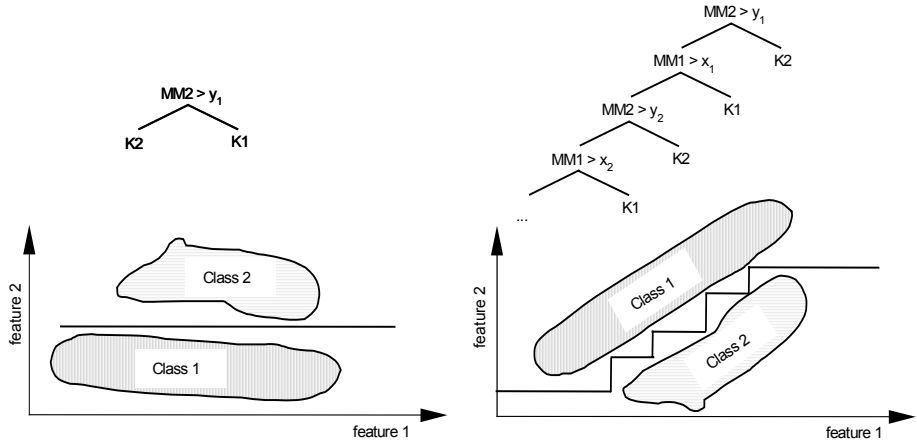


Fig. 18. Orthogonal and oblique class borders and corresponding decision trees

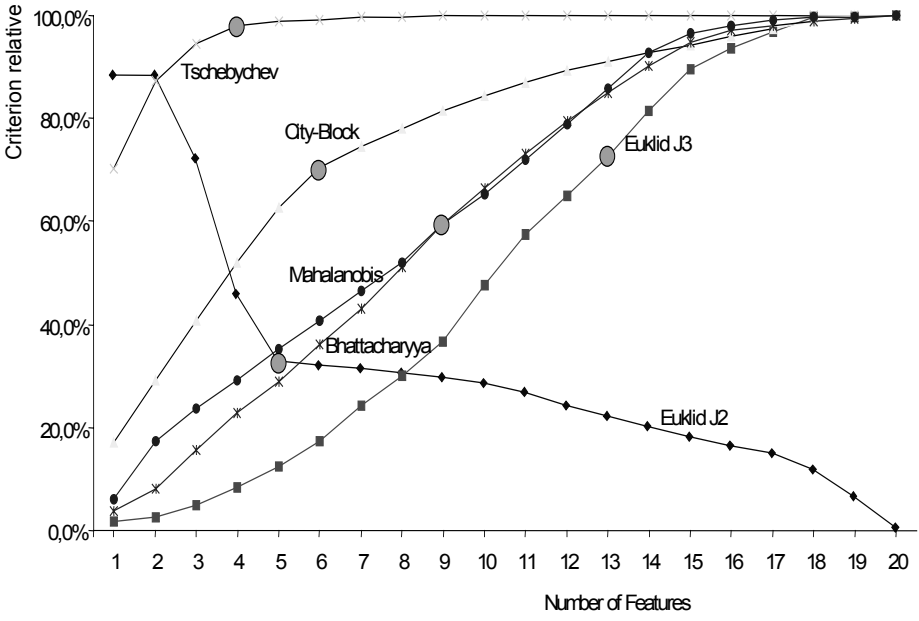


Fig. 19. Criteria of different filters as in [13] for an increasing number of features(SBS) related to 100% for 20 features of the pump problem. The dot indicates the feature subset actually chosen for testing.

As shown in figure 18 this approach favours features which require oblique borders in the feature space for class separation. Some weighting rules have been considered but none performed satisfyingly on the pump data

- Weighting of feature frequency with the number of data sets of the node

- Weighting of feature frequency with the number of correctly classified data sets of the node
- Weighting of feature frequency with the node level in the tree

6.2 Comparison of Filter Algorithms

The filtering of features for classification with See5 using a cost matrix is challenging because due to the cost matrix in See5 the optimisation goal differs between filter and classificator.

Several filter criteria, applied in a single-stage SFS, are plotted in figure 19. All criteria have a monotonous evolution, therefore the determination of the most suitable feature subset is arbitrary and has to be based on curvature or a similar criterion.

In addition a Bayes-classifier and a 1-Nearest-Neighbor-Classifier have been combined with a single stage SFS. The best feature subset determined with each filter was subjected to a parameter study in See5 (figure 20). Most filters lead to poor performing feature subsets. Only the Bhattacharyya-distance achieves similar results as the single-stage SFS wrapper with See5. A general suitability of probabilistic filters is refuted by the Mahalanobis-distance.

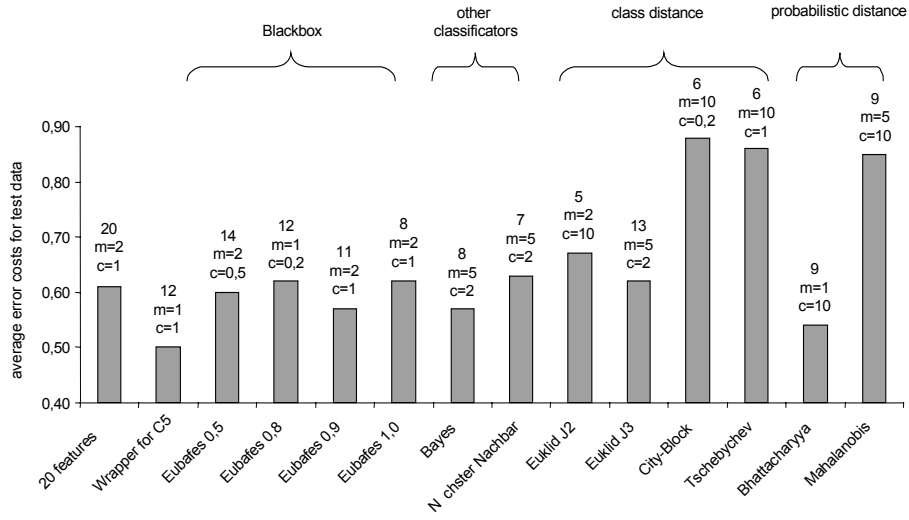


Fig. 20. Average costs for test data (mean value of 3 cross validations) obtained with the best See5-decision tree after a parameter optimisation with 4 values for Cases (m) and 7 values for Pruning (c).

7 Conclusion

Several feature selection algorithms have been compared on a real world machine learning problem. Due to its flexibility for modelling a complex relationship among

classes with cost matrices, See5 has been chosen as induction algorithm. Sequential forward and backward selection has been studied in depth. Using multistage approaches very effective feature subsets have been identified. A new genetic wrapper algorithm reduces computing time while increasing the number of nearly optimal subsets determined. As expected filter algorithms do perform worse than wrappers because of the particular optimisation goal introduced through the cost matrix. The application of filters only seems to be advisable with slowly converging classifiers such as neural networks, when computing time limits the usage of wrappers.

References

1. Cardie, C.: Using decision-trees to improve case-based learning; 10. Conf on Machine Learning; Wien; 1993
2. Devijer, P.; Kittler, J.: Pattern Recognition – A statistical approach; Prentice / Hall; 1982
3. Duda, R.; Hart, P.: Pattern classification and scene analysis; John Wiley & Sons; 1973
4. Kohavi, R.: Wrappers for performance enhancement and oblivious decison graphs; Dissertation; Stanford University; 1995
5. Lim, T.-S., Loh, W.-Y. and Shih, Y.-S.: A comparison of prediction accuracy, complexity, and training time of 33 old and new classification algorithms; Machine Leraning; preprint www.recursive-partitioning.com/mach1317.pdf; 1999
6. Martens et al: 'An initial comparison of a fuzzy neural classifier and a decision tree based classifier', Expert Systems with Applications (Pergamon) 15; 1998
7. Michie, D.; Spiegelhalter, D.; Taylor, C.: Machine Learning, Neural and Statistical Classification; Ellis Horwood; 1994
8. Perner, P.; Apte, C.: Empirical Evaluation of Feature Subset Selection Based on a Real World Data Set, In: D.A. Zighed, J. Komorowski, and J. Zytkow, Principles of Data Mining and Knowledge Discovery, Springer; 2000
9. Quinlan, J. R., Comparing connectionist and symbolic learning methods, Computational Learning Theory and Natural Learning Systems; Constraints and Prospects, ed. R. Rivest; MIT Press, 1994
10. Scherf, M.; Brauer, W.: Feature Selection by Means of a Feature Weighting Approach; Technical Report No FKI-221-97; Forschungsberichte Künstliche Intelligenz, Institut für Informatik, TU München; 1997
11. See5 / C5; Release 1.10; Quinlan, J. R. ;www.rulequest.com; 1999
12. Spath, D.; Hellmann, D. H.: Automatisches Lernen zur Störungsfrüherkennung bei ausfallkritischen Anlageelementen; Abschlußbericht zum DFG- Forschungsprojekt Sp 448/7-3 und He 2585/1-3; 1999
13. Rauber, T.: Tooldiag - Pattern Recognition Toolbox; Version 2.1; www.inf.ufes.br/~thomas/home/tooldiag; 1994

Automatic Identification of Diatoms Using Decision Forests

Stefan Fischer and Horst Bunke

Institute of Computer Science and Applied Mathematics
University of Bern, Switzerland
{fischer,bunke}@iam.unibe.ch

Abstract. A feature based identification scheme for microscopic images of diatoms is presented in this paper. Diatoms are unicellular algae found in water and other places wherever there is humidity and enough light for photo synthesis. The proposed automatic identification scheme follows a decision tree based classification approach. In this paper two different ensemble learning methods are evaluated and results are compared with those of single decision trees. As test sets two different diatom image databases are used. For each image in the databases general features like symmetry, geometric properties, moment invariants, and Fourier descriptors as well as diatom specific features like *striae* density and direction are computed.

1 Introduction

In previous studies [4] it turned out that decision trees learned on a diatom feature database are very specific to the training data. This phenomenon, known as overfitting, is a persistent problem in using decision trees for classification [9]. In existing decision tree based classification approaches a fully trained tree is often pruned to improve the generalization accuracy even if the error rate on the training data increases [15]. In the last decade multiple approaches have been studied to overcome this problem. Promising results were achieved using not only single classifiers but ensembles of multiple classifiers. In terms of decision trees they are often called decision forests. In such methods a set of classifiers is constructed and new samples are classified by taking a vote on the results of these classifiers. This strategy is based on the observation that, for example, decision tree classifiers can vary substantially when a small number of training samples are added or deleted from the training set. This instability affects not only the structure of the decision trees, but also the classification decisions made by the trees. This means that two runs of a decision tree induction algorithm on slightly different data sets will typically disagree on the classification of some test samples.

In the context of automatic identification of diatoms a decision forest based classification system can be used to identify unknown objects with a much higher generalization accuracy than a single decision tree learned on the same data set. In our approach general features are extracted from objects in microscopic images

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 173–183, 2001.

and stored as a data set. Based on such data sets decision trees are induced, and used afterwards for the identification of new objects.

In the following section conditions for constructing good ensembles are given and in Section 3 some existing methods for the construction of ensembles are reviewed. In Section 4 the databases used to evaluate our identification scheme and the test set-up are described. In Section 5 experimental results for single decision trees are shown and compared to those of bagging and the random subspace method. Finally conclusions are drawn in Section 6.

2 Conditions for Good Ensembles

In ensemble learning the individual decisions of classifiers from a set of classifiers are combined, for example, by majority vote, to classify new samples. Such ensembles are often much more accurate than the individual classifiers that make them up [3]. Nevertheless, there are some conditions which have to be fulfilled to get good results. A necessary and sufficient condition for an ensemble of classifiers to be more accurate than any of its individual members is that the classifiers are accurate and diverse [3]. An classifier is called accurate if it has an error rate which is better than random guessing on new samples. Two classifiers are called diverse if they make different errors on new samples. If these conditions are fulfilled the error rate of the classifier ensemble often decreases.

In general a learning algorithm can be viewed as searching a space $\mathcal{H}$ of hypotheses to identify the best hypothesis in the space. A statistical problem arises when the amount of training data available is too small compared to the size of the hypothesis space. Without sufficient data, the learning algorithm can find many different hypotheses in $\mathcal{H}$ that all give the same accuracy on the training data. This problem can be overcome by constructing an ensemble out of a set of classifiers. In this case the algorithm can average the votes and find a good approximation to the true target function f.

Many learning algorithms like decision trees work by performing a kind of local search that may get stuck in a local optimum. An ensemble constructed by running multiple local search processes with different starting conditions may provide a better approximation to the function f than any individual classifier.

3 Construction Methods

Various methods have been proposed for constructing ensembles. In general the intention to build more than one classifier is that each individual classifier should learn a different aspect of the problem. This can be satisfied, for example, by forcing each classifier to learn on different parts of the training set, but there are also other approaches available. In the following, examples from two different families of construction methods are reviewed.

In the first category of methods, ensembles are constructed by manipulating the training samples to generate multiple classifiers. In this case the learning algorithm is run several times, each time with a different subset of the training

set. This technique works especially well for unstable learning algorithms were the deletion or addition of one or more samples results in a different tree.

The most straightforward way to manipulate the training set is bagging [2]. In each run, a bootstrap replicate of the original training set is presented to the learning algorithm. Given a training set T of m samples, a replicate $T^{'}$ is constructed by drawing m samples uniformly with replacement from T.

Another training set sampling method is motivated by crossvalidation. In this method the training sets are constructed by leaving out disjoint subsets of the training data. An example is described in [14].

The third method for manipulating the training set is boosting [8]. A boosting algorithm maintains a set of weights over the original training set T and adjusts these weights after each classifier is learned by the base learning algorithm. In each iteration i, the learning algorithm is invoked to minimize the weighted error on the training set. The weighted error of the hypothesis h_i is computed and applied to update the weights on the training samples. Weights of samples that are misclassified by h_i are increased and weights of samples that are correctly classified are decreased. The final classifier is constructed by a weighted vote of the individual classifiers h_i. New training sets $T^{'}$ are constructed by drawing samples with replacement from T with a probability proportional to their individual weights.

In the second family of techniques for constructing multiple classifiers the set of features is manipulated. One example that belongs to this family is the random subspace method [9]. In this method multiple classifiers are generated by choosing subsets of the available features. An ensemble of classifiers is constructed by taking randomly half of the features to construct individual trees. If there are enough features available and if the features tend to be uncorrelated, the resulting ensemble classifier often has a higher accuracy than a single decision tree classifier [9].

In the following the performance of single decision trees in contrast to bagging and the random subspace method is evaluated on two different diatom databases.

4 Test Set-Up

The work reported in this paper has been done in the framework of a project which deals with the automatic identifcation of diatoms [1]. Diatoms are unicellular algae found in water and other places wherever there is humidity and enough light for photo synthesis. Diatom identification and classification has a number of applications in areas such as environmental monitoring, climate research and forensic medicine [20]. Example images of diatoms are shown in Fig. 1. As can be seen diatoms have various shapes and ornamentations. Also the size of diatoms varies over several orders of magnification, but most of them fall within the range of 10 to $100\mu m$ length. One of the great challenges in automatic diatom identification is the large number of classes involved. Experts estimate the number of described diatom species to be between $15,000$ and $20,000$, although this figure increases to approx. $100,000$ with the application of modern

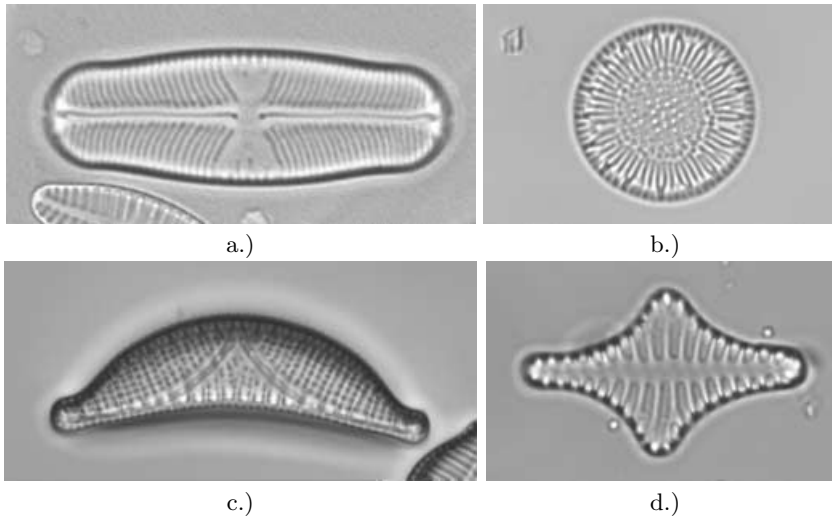


Fig. 1. Example images of diatoms. a.) *Sellaphora pupula*, b.) *Cyclotella radiosa*, c.) *Epithemia sorex*, d.) *Staurosirella leptostauron*

species concepts.Another 100,000 diatom species are estimated to be as yet undiscovered [12].

In this project several thousend images of diatoms have been captured and have been integrated into different databases. For the evaluation of decision tree based classifiers we chose two of those databases. The first database holds 120 images of diatoms which to date have been included in the same species of diatoms but actually represent a cluster of several tens of species[1]. (see Fig. 1a. for an example image). The images used here cover samples from 6 of these provisional species ("demes") which all have some specific characteristics of their shape. For example one class covers nearly rectangular ones while other classes hold more or less elliptical or blunt ones. For each class there are exactly 20 images in the database. Hence, the impact of the different classes to the induction of decision trees is balanced. In general this database is designed to analyze the performance of a classifier on samples which have nearly equal characteristics.

The second database holds images of 188 different diatoms which belong to 38 classes. There are at least 3 images available per class and in average there are nearly 5 images per class. The diatoms in this database vary not only in shape but also in texture such as the images in Fig. 1b.-d.). This database is used to analyze the performance of a classifier on a diversity of diatoms.

In contrast to earlier works [13,19] not only features of the shape are used to describe the different characteristics of diatoms, but also features of the ornamentation. The whole set of features used in the classifiers described in this

[1] In terms of biologists diatoms are hierarchical classified in *genus*, *species*, *subspecies* and so forth, but in this paper we'll use the term *class* in the pattern recognition sense.

Table 1. Types of symmetry which are used to destinguish different classes of diatoms

Class of symmetry	Description
0	one symmetry axis (principal axis)
1	one symmetry axis (orthogonal to the principal axis)
2	two symmetry axes
3	three symmetry axes
4	four or more symmetry axes for circular ones

paper include invariant moments [7,10], Fourier descriptors, simple scalar shape descriptors, symmetry descriptors, geometric properties as well as diatom specific features like *striae* density and direction are used. For a description of the feature extraction procedure see [5].

As scalar shape descriptors triangularity, ellipticity [17], rectangularity, circularity, and compactness [18] are used. Even these simple descriptors have shown good descriminating ability. As these descriptors correspond well with human intuitve shape perception, they make it much easier for a human expert to interpret the decisions made by the classifier.

A widely used property in the identification of diatoms is their type of symmetry. There are forms which have one, two or even more symmetry axes. In general we distinguish between five different types of symmetry as shown in Table 1. How the class of symmetry can be used in the identification process of diatoms is described in [6].

An important feature of the ornamentation of diatoms is the *striae* which appear as stripes on both sides of the middle axis. For example, the diatom in Fig. 1a.) has stripes, while the one in Fig. 1b.) has none. The stripe density and the mean direction are known for most classes of diatoms, and remain constant during the entire life cycle of a diatom. This makes them a very important feature for the description of certain types of diatoms independent of their shape.

In Table 2 all feature which are available to the induction process are listed. In total there are 149 features used, most of which are Fourier descriptors. In the first column of Table 2 the group of the feature is specified and in the second column the names of the single features are listed. In the last column the number of features per group is given.

With this set of features decision trees are build using the C4.5 algorithm [16].

5 Experimental Results

The proposed decision tree based classification methods will be an integral part of an automatic diatom identification system. The goal of the system is to assist an untrained human in the identification of a wide range of diatoms. Instead of presenting a final identification of an unknown object, a list of possible matches

Table 2. Features used for the induction of decision trees

Group	Feature	Number
Moment invariants	moment invariants proposed by Hu (7) moment invariants proposed by Flusser (4)	11
Fourier descriptors	normalized fourier descriptors	126
Scalar shape descriptors	rectangularity, triangularity, circularity, ellipticity, compactness	5
Symmetry	class of symmetry	1
Geometric properties	length, width, length/width-ratio, size	4
Diatom specific features	*striae* density, direction	2

will be given to the user. From such a list the user can decide to which of the suggested classes the unknown object belongs to.

To evaluate the performance of decision tree classifiers on the two diatom databases introduced in the previous section, three tests were performed. In the first test single decison trees were build. In the second test 20 bootstrap replicates of the training set were build following the bagging approach described in [2]. As classification result the majority class of all 20 classifiers is used. In the last test instead of building replications of the training set, 100 random subsets of the available feature set were choosen as described in [9]. Each of the subsets contained exactly half of the available features. The final classification is the majority class of all classifiers, again.

All results were validated following the leave one out approach. This means each sample in the databases was once used for testing and all other samples were used for training. This procedure was repeated until each sample was used exactly once for testing.

The results on the first database are visualized in Fig. 2. In each of the diagrams the results of all three experiments are displayed. The recognition rate achived by single decision trees is drawn as a dotted line, the rate for bagging as a solid line, and for the random subspace method as a dashed line. On the y-axis the recognition rate is given and on the x-axis the highest rank taken into regard. Thus, for example, rank 2 in the chart represents the accumulated recognition rate for all samples whose real class is detected as the first or second possible class by the decision tree based classifier.

As can be seen in Fig. 2a.) the recognition rate using single decision trees starts with slightly more than 90 percent and reaches its maximum of 93.33% at the third rank. In total there are 8 samples where the right class was not among the first three ranks.

For bagging the recognition rate for the first rank is 92.5% and the maximum is reached already on the second rank with 99.17%. For this approach there is still one sample which can not be assigned to the right class.

For the random subspace method the initial recognition rate is slightly higher than for the other two methods. Now on the third rank all samples are assigned to the right class and therefore a recognition rate of 100% is obtained.

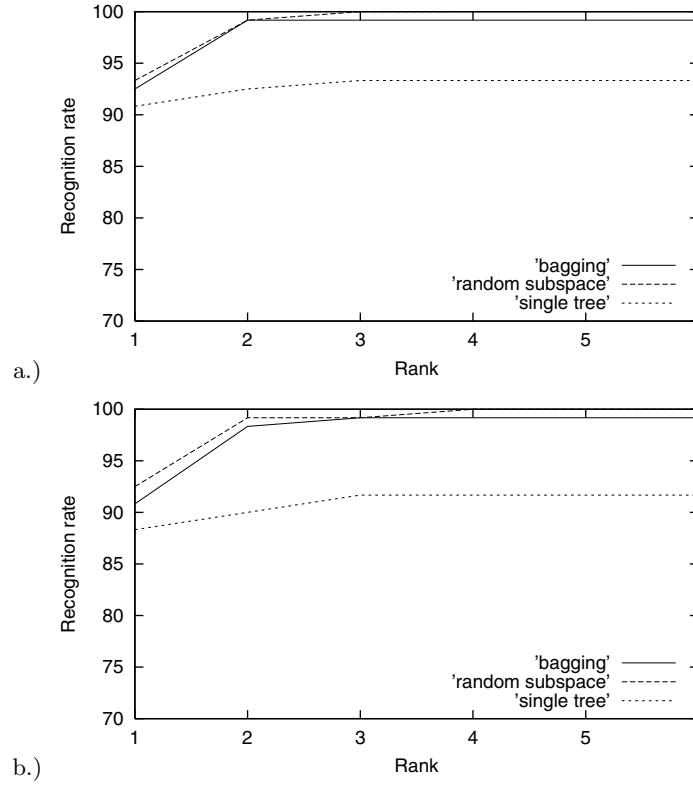


Fig. 2. Recognition rates on the first database for single tree, bagging, and randomization. a.) complete feature set, b.) reduced feature set.

These results show that both ensemble learning methods have much better recognition rates than single decision trees. The reason for this is that there is no feature available which allows to discriminate all diatoms of the different classes perfectly. Thus, the decision tree induction algorithm has to make arbitrary decisions on the available features and therefore the classifier is instable and the use of ensembles results in this case in better recognition rates.

To evaluate the influence of the number of features we made a second run with a reduced feature set. While for the first run the total set of 149 features including 126 Fourier descriptors was available to the decision tree induction process, the feature set was restricted in the second run to "human interpretable" features. This can be very important if, for example, a diatomist wants to judge the decisions made by an automatic procedure. Thus, from the complete set of features used during the first run we have removed the Fourier descriptors and the moment invariants which are difficult to interpret. Additionally we decided to remove the measures for triangularity and compactness because their computation is closely related to other features. In Table 3 the features of the reduced feature set are listed.

As can be seen in Fig. 2b.) the recognition rates are nearly the same as in the first test. In general the curves are a bit flatter and the maximum is reached on a higher rank but for bagging and the random subspace method still the same maximal recognition rate is reached.

Now the results are validated one the second database holding images of 38 different classes. Once again the complete feature set was used in the first run. As can be seen in Fig. 3a.) the recognition rate using single decision trees now is much lower than in the previous test. This is an indication for the lack of training data in the decision tree induction process. Clearly, in a situation were the recognition rate is still much better than random guessing we can expect that ensemble methods will lead to better results than any single classifier can do. This behavior is reflected in the curves for bagging and randomization in Fig. 3a.).

Starting from the first rank the recognition rate for bagging and randomization is substantially higher (77.13% resp. 79.26%) than that using single decision trees (65.53%). For the two ensemble approaches the recognition rate rises continuously and with the fifth rank a rate of 95.74% resp. 96.81% is achieved. Even if the recognition rate increases for higher ranks none of the considered approaches can classify all samples of this much more complex training set correctly.

An analysis of the misclassified samples shows different reasons for this problem. In Fig. 4 two example diatoms of the second database are shown which are often assigned to a wrong class. For example the diatom in Fig. 4a.) has nearly the same shape as diatoms from other classes. At the same time it is different from other diatoms of the same class with respect to shape. The latter can be due to a variety of factors including genetic and environmental influences, but most morphological variation is due to changes occurring over the diatom life cycle. Even for a human expert it seems to be difficult to identify this kind of diatom correctly [11].

Another example of misclassification is shown in Fig. 4b.). This kind of diatom has a very special internal structure that differs significantly from other diatoms (see for example Fig. 1a.-d.). At the moment there is no feature available which is capable to describe this structure.

Table 3. Reduced feature set used to evaluate the influence of features

Group	Feature	Number
Scalar shape descriptors	rectangularity, triangularity, circularity	3
Symmetry	class of symmetry	1
Geometric properties	length, width, length/width-ratio, size	4
Diatom specific features	*striae* density, direction	2

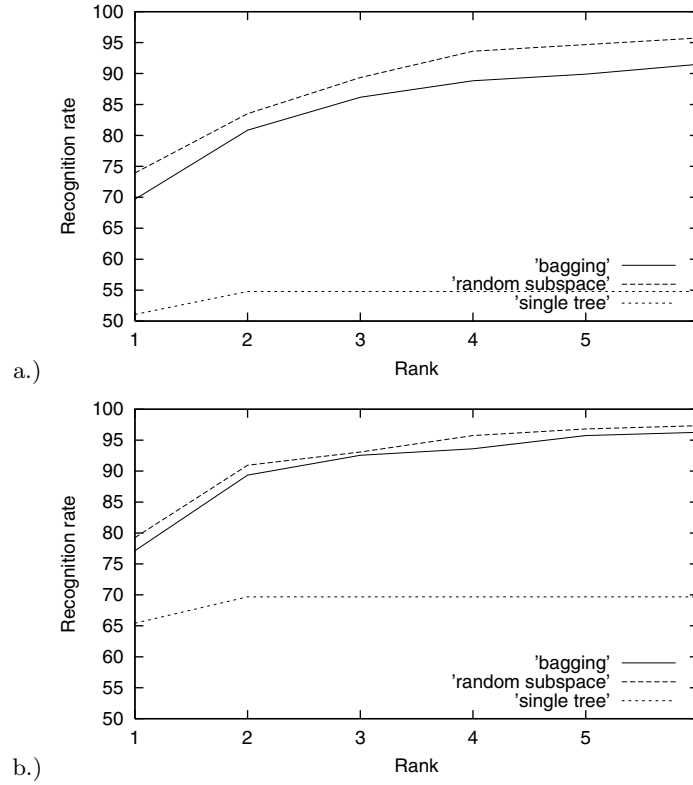
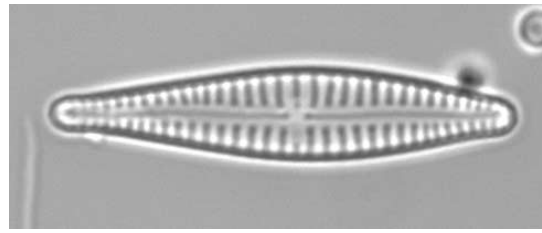


Fig. 3. Recognition rates on the second database for single tree, bagging, and randomization. a.) complete feature set, b.) reduced feature set.

6 Conclusion

In this paper we have presented three different methods which can be used in a decision tree based classification system. We have compared the results of single decision trees with bagging and the random subspace method. The results on two different diatom image databases have shown that much better results are obtained by using decision forests than single decision trees. In general the recognition rates for bagging and the random subspace method are very close to each other, but the random subspace method slightly outperforms bagging in all of the tests. Even on a very complex database holding images of 38 different classes of diatoms recognition rates of more than 95% were achieved if the first five rank are taken into regard. The misclassification of certain images is the result of the variability and/or lack of special characteristics of diatoms as well as the lack of specific features to capture typical characteristics of single groups of objects. In the future the goal of our work will be to further improve the classification performance of our automatic identification system, although the

a.)



b.)

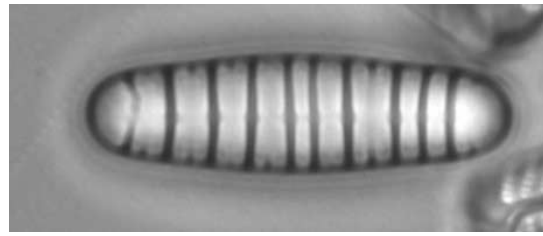


Fig. 4. Two example images from the second database which are misclassified by all approaches. a.) *Gomphonema parvulum*, b.) *Diatoma vulgaris*.

recognition rates are already impressive compared with those of earlier works where always single species were regarded.

Acknowledgment

The work has been done in the framework of the EU-sponsored Marine Science and Technology Program (MAST-III), under contract no. MAS3-CT97-0122. We thank our project partners Micha Bayer and Stephen Droop from Royal Botanic Garden Edinburgh and Steve Juggins and co-workers at Newcastle University for preparing the images in the ADIAC image database and for useful discussions and hints.

References

1. Automatic Diatom Identification And Classification. Project home page: http://www.ualg.pt/adiac/.
2. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
3. T. G. Dietterich. Ensemble methods in machine learning. In J. Kittler and F. Roli, editors, *First International Workshop on Multiple Classifier Systems*, Lecture Notes in Computer Science, pages 1–15. Springer Verlag, New York, 2000.
4. S. Fischer, M. Binkert, and H. Bunke. Feature based retrieval of diatoms in an image database using decision trees. In *ACIVS 2000*, pages 67–72, Baden-Baden, Germany, August 2000.
5. S. Fischer, M. Binkert, and H. Bunke. Feature based retrieval of diatoms in an image database using decision trees. Technical Report IAM-00-001, Institute of Computer Science and Applied Mathematics, University of Bern, Switzerland, 2000.

6. S. Fischer, M. Binkert, and H. Bunke. Symmetry based indexing of diatoms in an image database. In *Proceedings of the ICPR 2000*, volume 2, pages 899–902, Barcelona, Spain, September 2000.
7. J. Flusser and T. Suk. Pattern recognition by affine moment invariants. *Pattern Recognition*, 26(1):167–174, 1993.
8. Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *Machine Learning: Proceedings of the Thirteenth International Conference*, pages 148–156, Bari, Italy, 1996. Morgan Kaufmann.
9. T. K. Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, August 1998.
10. M. Hu. Visual pattern recognition by moment invariants. *IEEE Trans. Information Theory*, 8(2):179–187, February 1962.
11. K. Krammer and H. Lange-Bertalot. Bacillariophyceae. In H. Ettl, J. Gerloff, H. Heynig, and D. Mollenhauer, editors, *Süsswasserflora von Mitteleuropa (in German)*. Gustav Fischer Verlag, Stuttgart, 1986.
12. D. G. Mann, and S. J. M. Droop. Biodiversity, biogeography and conservation of diatoms. Hydrobiologia 336:19-32, 1996.
13. D. Mou, and E. F. Stoermer. Separating Tabellaria (Bacillariophyceae) shape groups: A large sample approach based on Fourier descriptor analysis. *Journal of Phycology*, 28:386-395, 1992.
14. B. Parmanto, P. W. Munro, and H. R. Doyle. Reducing variance of committee prediction with resampling techniques. *Connection Science*, 8(3&4):405–425, 1996.
15. J. R. Quinlan. Bagging, boosting, and C4.5. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 725–730, Cambridge, MA, 1996. AAAI Press/MIT Press.
16. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, USA, 1993.
17. P. L. Rosin. Measuring shape: Ellipticity, rectangularity, and triangularity. In *Proceedings of the ICPR 2000*, volume 2, pages 952–955, Barcelona, Spain, September 2000.
18. M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Brooks/Cole, 2. edition, 1999.
19. E. F. Stoermer, and T. B. Ladewski. Quantitative analysis of shape variations in type and modern populations of Gomphoneis herculeana. *Nova Hedwigia Beihefte*, 73:347-373, 1982.
20. E. F. Stoermer and J. P. Smol, editors. *The Diatoms: Applications for the Environmental and Earth Science*. Cambridge University Press, 1999.

Validation of Text Clustering Based on Document Contents

Jarmo Toivonen[1], Ari Visa[1], Tomi Vesanen[1],
Barbro Back[2], and Hannu Vanharanta[3]

[1] Tampere University of Technology
P.O. Box 553, FIN-33101 Tampere, Finland
{Jarmo.Toivonen, Ari.Visa, Tomi.Vesanen}@tut.fi
[2] Åbo Akademi University
Lemminkäisenkatu 14 A, FIN-20520 Turku, Finland
Barbro.Back@abo.fi
[3] Pori School of Technology and Economics
P.O. Box 300, FIN-28101 Pori, Finland
Hannu.Vanharanta@pori.tut.fi

Abstract. In this paper some results of a new text clustering methodology are presented. A prototype is an interesting document or a part of an extracted, interesting text. The given prototype is matched with the existing document database or the monitored document flow. Our claim is that the new methodology is capable of automatic content-based clustering using the information of the document. To verify this hypothesis an experiment was designed with the Bible. Four different translations, one Greek, one Latin, and two Finnish translations from years 1933/38 and 1992 were selected as test text material. Validation experiments were performed with a designed prototype version of the software application.

1 Introduction

Nowadays a large amount of information is stored in Intranet, Internet or in databases. Customer comments and communications, trade publications, research reports and competitor web sites are just a few examples of available electronic data. Everyone needs a solution for handling the large volume of unstructured information they confront each day. It is extremely important to find the desired information but the information needs varies. There are needs to document retrieval, document filtering, or text mining. These methods are usually based natural language processing. There are techniques that are based on index terms [4] but the index term list is fixed. There are techniques that are based on vector space models [8,7] but these techniques miss the information of co-occurrences of words. There are techniques that are capable to consider the co-occurrences of words, as latent semantic analysis [3,6] but they are computationally heavy. A common approach to topic detection and tracking is usage of keywords, especially in context of Dewey Decimal Classification [2,1]. This approach is based on assumption that the keywords given by the authors characterise the text well. This might be true but then one neglects the accuracy.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 184–195, 2001.

More accurate method is to use all the words of a document and the frequency distribution of words. Now the comparison of frequency distributions is a complicated task. There are theories that the rare words in the histograms distinguish documents [5]. Our approach utilises this idea but in a peculiar way. The idea is expanded also to sentence and paragraph levels.

In this paper we represent our methodology briefly and concentrate on tests of content based topic classification. It is something that is highly attractive in text mining. The evolution of the methodology has been earlier discussed in several publications [11,9,10]. In the second chapter the applied methodology is described. In the third chapter the designed experiments are described and the validation results are reported. Finally, the methodology and the results are discussed.

2 Methodology

The methodology is briefly based on word, sentence, and paragraph level processing. The original text is first preprocessed, extra spaces and carriage returns are omitted, etc. The filtered text is next translated into a suitable form for encoding purposes. The encoding of words is a wide subject and there are several approaches for doing it:
1) The word is recognised and replaced with a code. This approach is sensitive to new words.
2) The succeeding words are replaced with a code. This method is language sensitive.
3) Each word is analysed character by character and based on the characters a key entry to a code table is calculated. This approach is sensitive to capital letters and conjugation if the code table is not arranged in a special way.

We chose the last alternative, because it is accurate and suitable for statistical analysis. A word w is transformed into a number in the following manner:

$$y = \sum_{i=0}^{L-1} k^i * c_{L-i} \tag{1}$$

where L is the length of the character string (the word), c_i is the ASCII value of a character within a word w, and k is a constant.

Example: word is "**c a t**".

$$y = k^2 * ascii(c) + k * ascii(a) + ascii(t) \tag{2}$$

The encoding algorithm makes a different number for each different word, only the same word can have an equal number. After each word has been converted to a code number we set minimum and maximum values to words, and look the distribution of words' code numbers. Now one tries to estimate the distribution of the code numbers. Weibull distribution is selected to represent the distribution. Other distributions, e.g. Gamma distribution, are also possible. However,

it would be advategous, if the selected distribution had only a few parameters and it matched the observed distribution as well as possible.

In the training phase the range between the minimum and the maximum values of words' code numbers is divided to N_w logarithmically equal bins. The frequency count of words belonging to each bin is calculated. The bins' counts are divided with the number of all words. Then the best Weibull distribution corresponding to the data must be determined. Weibull distribution is compared with distribution by examining both distributions' cumulative distribution. Weibull's Cumulative Distribution Function is calcu lated by:

$$CDF = 1 - e^{(((-2.6*log(y/y_{max}))^b)*a)} \tag{3}$$

There are two parameters that can be changed in Weibull's CDF formula: a and b. A set of Weibull distributions are calculated with all the possible combinations of a's and b's using a selected precision. The possible values for the coefficients are restricted between suitable minimum and maximum values. The cumulative code number distribution and Weibull's cumulative distribution are compared in the smallest square sum sense.

In the testing phase the best Weibull distribution is found and it is now divided to N_w equal size bins. The size of every bin is $1/N_w$. Every word belongs now to a bin that can be found using the code number and the best fitting Weibull distribution. Using this type of quantisation the word can now be presented as the number of the bin that it belongs to. Due to the selected coding method the resolution will be the best where the words are most typical to text (usually 2-5 length words). Rare words (usually long words) are not so accurately separated from each other. Similarly on the sentence level every sentence has to be converted to a number. First every word in a sentence is changed to a bin number in the same way we did with words earlier.

Example:

I have a cat .
bn_0 bn_1 bn_2 bn_3 bn_4

where bn_i = bin number of the word i.

The whole encoded sentence is now considered as a sampled signal. The signal is next Fourier transformed. Since the sentences of the text contain different numbers of words, the sentence vectors' lengths differ. Here we use the Discrete Fourier Transform (DFT) to transform the sentence vectors. We do not consider all the coefficients. The input for the DFT is $(bn_0, bn_1, ..., bn_n)$. DFT's outputs are coefficients B_0 to B_n. The second coefficient B_1 is selected to be the number that describes the sentence. The reason why the B_1 component is selected is that in the experiments it has been observed that B_0 is too much effected by the sentences' length.

After every sentence has been converted to numbers, a cumulative distribution is created from the sentence data set in the same way as on the word level. Now the range between the minimum and the maximum value of the sentence

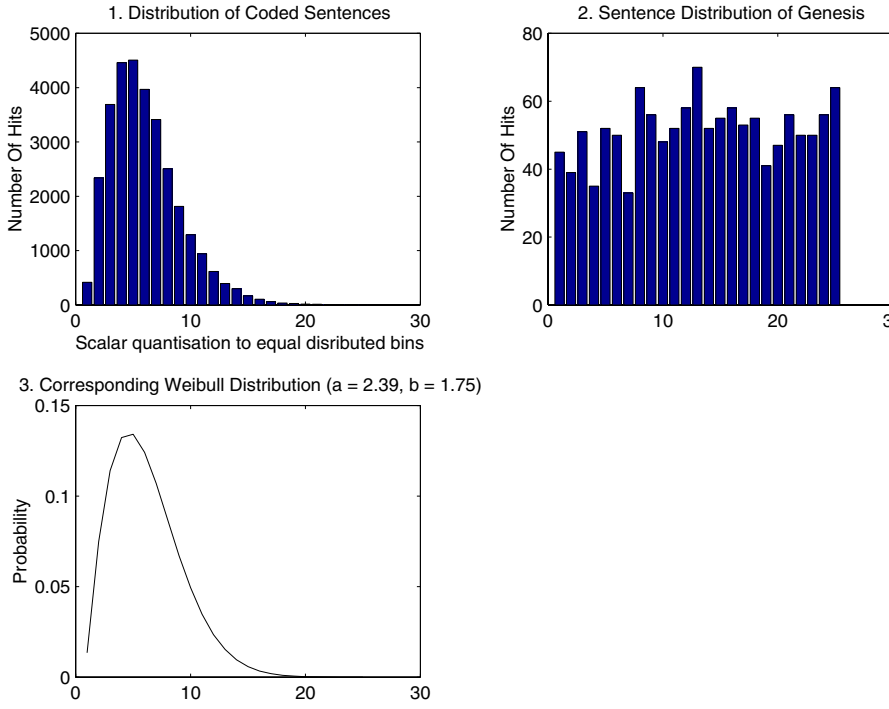


Fig. 1. Example of a sentence quantisation process.

code numbers are divided to N_s equal size bins. The frequency count of sentences belonging to each bin is calculated and the bins' counts are divided with the number of all sentences. The best Weibull distribution corresponding to the sentence data is found using the cumulative distribution of both distributions. Now the best distribution can be used in the quantisation of sentences. An example of a sentence distribution and a corresponding best Weibull distribution are illustrated in Fig. 1, subplots 1 and 3. In these examples the number of the bins N_s is 25. On the paragraph level the methods are similar. The paragraphs of the document are first converted to vectors using the code numbers of the sentences. The vectors are Fourier transformed and the coefficient B_1 is chosen to represent the paragraph. After the best Weibull distribution corresponding to the paragraph data is found it can be used in the quantisation of paragraphs.

When examining the single text documents, we create histograms of the documents' word, sentence, and paragraph code numbers according to the corresponding value of quantisation. On the word level the filtered text from a single document is encoded word by word. Each word code number is quantised using word quantisation created with all the words of the data base. The right quantisation value is determined, an accumulator corresponding to the value is increased, and thus a word histogram A_w is created. The histogram A_w consist-

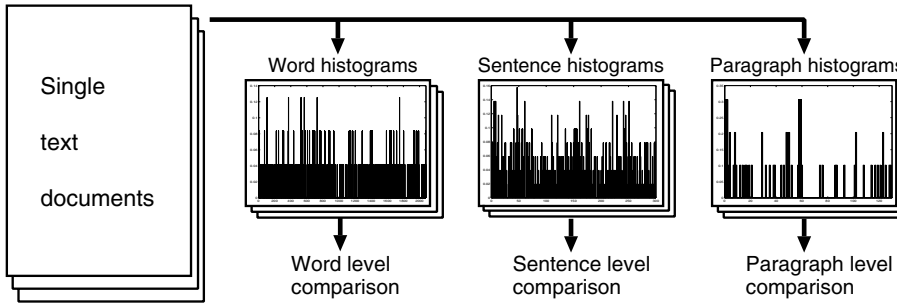


Fig. 2. The process of comparing and analysing documents based on the extracted histograms on different levels.

ing of N_w bins is finally normalised by the total word count of the document. On the sentence and the paragraph levels the histogram creation process is similar. The single document is encoded to sentence and paragraph code numbers and the hits according to the corresponding place in the quantisation are collected in histograms A_s and A_p. An example of a sentence histogram is illustrated in Fig. 1, subplot 2. With the histograms from all the documents in the database we can compare and analyse the single documents' text on the word, sentence, and paragraph levels. The histogram creation and comparison processes are illustrated in Fig. 2. Note, that it is not necessary to know anything from the actual text document to do this. It is sufficient to give one document as a prototype. The methodology gives the user all the similar documents, gives a number to the difference, or clusters similar documents.

3 Experiments

Our assumption is that the textual clustering depends on given factors:

$$Text\ Clustering = Message + Style + Language + Method \tag{4}$$

To find out which of the mentioned factors is most powerful an experiment was designed. In the tests we kept the method the same all the time and varied the style, language, and message. It was important to find a text that is carefully translated into another language. In translation it is important, at least, to keep the message the same even though the form depends on the language. The Bible was selected to meet the demands. The translations used were the Westcott-Hort translation in Greek and the translations from years 1933 (the Old Testament), 1938 (the New Testament), and 1992 (whole Bible) in Finnish. In the first test we also used a Latin version of the Bible for comparison. The Latin translation was Jerome's translation (Vulgate) from years 382-405. As measures precision and recall were used. The idea was to select a recall window of closest matches to 10 and to compare all the books in the Bible. The size of the histograms, for the word level was 2080, for the sentence level 25, and for the paragraph level

10. The word, the sentence and paragraph level histograms were created based on the whole text of the Bibles. Euclidean distance was used in the comparisons of the histograms.

In the first experiment the capability of the methodology to separate documents on a coarse level was examined. We know that the Old and the New Testament books differ and expected to see a difference in the first test. Every book was one by one taken as a prototype document, and ten closest matches were examined. Note, that the order within the window is not considered, only the co-occurrences. The number of books in the window that matched with other books in the Old Testament, respectively in the New Testament are reported for four translations in Tables 1, 2, 3, 4. For example, for the Genesis (book number 1) in Greek, we see that on the word and the paragraph level there are eight Old Testament books among the ten closest books, and on the sentence level six. At the word level on average eight books from ten were from the assumed class. At the sentence level there was more variety, from five to six books were from the assumed class. At the paragraph level on average five books from ten were from the assumed class.

The second experiment looks at the differences between the languages or translations and the style in more detail. Now each pair of two different translations are selected from the group of three translations. Again the ten closest matches are examined. The results are presented in Tables 5, 6, 7. Now for example for the Genesis there are seven same books among the ten closest in Greek and Finnish 1933/1938 translations on the word level, five on the sentence level and six on the paragraph level. It can be observed that the differences concentrate on word and sentence levels. At paragraph level the structure of the text wins over the style.

In the third experiment the effect of the message is studied. We know that in the Bible the books can be divided into groups based on similarity of their contents. Two this kind of distinct groups are the books 18-22 (Job, Psalms, Proverbs, Ecclesiastes, Song of Solomon) and the books number 40-44 (the gospels by Matthew, Mark, Luke, and John and the Acts). By examining these groups we try to find out how well our methodology is capable of clustering these texts, keeping in mind the style and language effects. In the experiments recall window size five is used. The books of the same group for each five prototype book among the five closest books are counted. The results are presented in tables 8, 9, 10. The reason why recall window is now five is that there are no more than five books in both test sets. The meaning of the text plays an important role to the clustering result. The evidences to this conclusion are strong at the word and sentence levels. At the paragraph level the structural aspects start to play in.

4 Discussion

The main idea is to test the ability to find similar contents. One of our basic assumptions is that within a specific field, for instance in law or business, the

ambiguities of words will not disturb significantly. Our experiments are based on the model that the content of a document is described by the message, the language, and the style. That was the reason why the Bible was selected as test material. We know that the translations have been done very carefully, at least at the information level.

The influence of language and the style is eliminated by using four different translations of the Bible. First a simple test was designed: the task was to distinguish between the Old Testament and the New Testament. The search was done by taking one book as a prototype and all similar books to that book were searched. Ten closest matches were displayed and all the books were checked. The results were similar from language to language. At the word level on average eight books from ten were from the assumed class. At the sentence level there was more variety, from five to six books were from the assumed class. At the paragraph level on average five books from ten were from the assumed class. The influence of style was studied more based on different versions of the Bible in one specific language. It can be observed that the differences concentrate on word and sentence levels. At paragraph level the structure of the text wins over the style. Finally, at the message level it seems that the meaning of the text plays an important role to the clustering result. The evidences to this conclusion are strong at the word and sentence levels. At the paragraph level the structural aspects start to play in. In general one should note that the numbers given in the tables should be related to corresponding numbers calculated with random samples. The observed numbers are several magnitudes higher than the calculated ones.

It seems that a methodology capable of content-based filtering has been developed. The presented methodology makes it possible to search text documents in a different way than by using keywords. The methodology can easily be adapted to new fields by training.

Acknowledgments

The financial support of TEKES (grant number 40943/99) is gratefully acknowledged.

References

1. M. Dewey. *A Classification and subject index for cataloguing and arranging the books and pamphlets of a library.* Case, Lockwood & Brainard Co., Amherst, MA, USA, 1876.
2. M. Dewey. Catalogs and Cataloguing: A Decimal Classification and Subject Index. In *U.S. Bureau of Education Special Report on Public Libraries Part I*, pages 623–648. U.S.G.P.O., Washington DC, USA, 1876.
3. F. C. Gey. Information Retrieval: Theory, Application, Evaluation. In *Tutorial at HICSS-33, Hawaii International Conference on System Sciences (CD-ROM)*, Maui, Hawaii, USA, Jan. 4–7 2000.

4. T. Lahtinen. *Automatic indexing: an approach using an index term corpus and combining linguistic and statistical methods.* PhD thesis, Department of General Linguistics, University of Helsinki, Finland, 2000.
5. C. D. Manning and H. Schütze. *Foundations of Statistical Natural Language Processing.* The MIT Press, Cambridge, Massachusetts, 1999.
6. D. W. Oard and G. Marchionini. A conceptual framework for text filtering. Technical Report CS-TR3643, University of Maryland, May 1996.
7. G. Salton. *Automatic Text Processing.* Addison-Wesley, 1989.
8. G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.
9. A. Visa, J. Toivonen, S. Autio, J. Mäkinen, H. Vanharanta, and B. Back. Data Mining of Text as a Tool in Authorship Attribution. In B. V. Dasarathy, editor, *Proceedings of AeroSense 2001, SPIE 15th Annual International Symposium on Aerospace/Defense Sensing, Simulation and Controls. Data Mining and Knowledge Discovery: Theory, Tools, and Technology III*, volume 4384, Orlando, Florida, USA, April 16–20 2001.
10. A. Visa, J. Toivonen, B. Back, and H. Vanharanta. Improvements on a Knowledge Discovery Methodology for Text Documents. In *Proceedings of SSGRR 2000 – International Conference on Advances in Infrastructure for Electronic Business, Science, and Education on the Internet*, L'Aquila, Rome, Italy, July 31–August 6 2000. (CD–ROM).
11. A. Visa, J. Toivonen, H. Vanharanta, and B. Back. Prototype Matching – Finding Meaning in the Books of the Bible. In J. Ralph H. Sprague, editor, *Proceedings of the Thirty-Fourth Annual Hawaii International Conference on System Sciences (HICSS-34)*, Maui, Hawaii, USA, January 3–6 2001. (CD–ROM).

Table 1. Number of books from the Old testament, respectively the New testament, among ten closest matches in the Greek translation.

	Old Testament, book number																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	8	9	10	10	9	10	9	10	10	10	10	9	10	10	10	10	6
Sentence	6	6	6	5	9	8	10	5	9	9	8	8	6	8	6	7	8
Paragraph	8	8	8	9	8	8	5	6	8	8	8	7	9	8	9	9	5

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	2	5	1	8	4	9	9	8	10	9	9	10	10	7	10	9	10	8	10	9	10	10
Sentence	4	5	5	6	5	8	8	8	6	9	7	7	7	8	9	8	9	9	7	8	9	7
Paragraph	6	8	9	8	6	8	8	8	6	8	7	5	6	9	6	5	6	10	6	6	7	7

	New Testament, book number																											
	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	6	2	4	7	8	10	9	10	10	8	6	9	9	10	10	10	10	9	8	9	10	10	8	8	8	10	0
Sentence	4	3	3	4	3	4	4	3	4	5	4	2	3	3	2	3	5	5	2	5	3	2	2	7	5	2	1
Paragraph	0	2	1	4	0	4	2	3	3	5	0	2	3	5	4	6	5	4	5	4	6	2	4	2	4	4	4

	Old Testament average	**New Testament average**	**Total average**
Word	8.64	8.07	8.41
Sentence	7.26	3.44	5.70
Paragraph	7.33	3.26	5.67

Table 2. Number of books from the Old testament, respectively the New testament, among ten closest matches in the Finnish 1933/1938 translation.

	Old Testament, book number																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	10	10	10	10	10	10	10	8	10	10	10	10	10	10	9	10	10
Sentence	6	7	10	9	8	7	9	9	7	8	7	10	7	10	9	6	7
Paragraph	4	9	7	4	8	5	8	9	7	6	8	7	4	9	5	6	6

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	10	10	9	8	8	10	10	8	10	9	10	9	10	10	10	10	9	10	10	10	10	10
Sentence	6	5	7	4	7	6	9	4	9	9	6	5	9	6	3	5	4	5	7	6	9	7
Paragraph	9	9	8	7	7	8	8	6	9	8	5	6	8	6	9	8	5	7	7	7	8	5

	New Testament, book number																										
	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	5	5	4	8	4	10	10	10	10	10	5	10	10	10	10	10	10	6	9	10	10	9	10	10	10	7	3
Sentence	4	4	3	4	1	3	4	2	3	2	4	5	4	3	4	5	1	3	1	5	4	2	4	3	2	3	0
Paragraph	5	3	4	4	2	4	4	4	2	3	5	4	4	3	3	3	4	4	1	2	5	4	2	3	3	4	2

	Old Testament average	**New Testament average**	**Total average**
Word	9.67	8.33	9.12
Sentence	7.03	3.07	5.41
Paragraph	6.97	3.37	5.50

Table 3. Number of books from the Old testament, respectively the New testament, among ten closest matches in the Finnish 1992 translation.

	Old Testament, book number																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	6	9	4	9	4	8	6	6	6	6	7	6	8	8	6	8	8
Sentence	4	6	7	7	8	9	10	8	9	9	8	9	5	7	6	7	9
Paragraph	8	6	9	7	8	8	7	9	8	8	7	7	6	7	6	10	8

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	7	10	5	3	10	9	9	10	7	8	9	9	8	10	8	10	10	9	10	10	8	8
Sentence	5	5	7	4	6	4	9	5	7	8	4	5	8	9	8	4	5	8	4	5	9	9
Paragraph	8	8	6	7	6	10	7	7	8	6	7	5	8	4	7	8	6	7	5	7	7	7

	New Testament, book number																										
	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	4	4	4	3	2	7	6	9	7	10	4	9	9	9	9	7	8	8	7	8	9	8	8	8	4	7	3
Sentence	4	5	3	6	4	5	7	5	3	5	4	7	6	7	6	7	7	7	6	4	2	4	6	5	8	1	1
Paragraph	3	3	3	2	2	3	1	3	4	5	3	4	4	3	2	1	6	1	3	1	4	3	3	3	6	6	2

	Old Testament average	**New Testament average**	**Total average**
Word	7.74	6.70	7.32
Sentence	6.82	5.00	6.08
Paragraph	7.18	3.11	5.52

Table 4. Number of books from the Old testament, respectively the New testament, among ten closest matches in the Latin translation.

Old Testament, book number

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	8	10	9	8	10	10	9	7	8	10	10	10	10	10	10	9	9
Sentence	5	6	6	5	6	9	9	10	9	8	8	10	5	9	9	8	9
Paragraph	8	8	6	7	9	9	5	5	6	6	2	9	5	8	9	7	8

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	9	10	5	9	10	10	10	7	9	9	9	10	10	10	8	10	10	10	10	10	9	9
Sentence	4	4	4	5	5	6	9	6	9	9	5	8	7	3	5	8	7	5	9	5	9	8
Paragraph	7	10	5	8	7	9	8	7	7	6	7	5	7	6	6	7	5	5	7	7	7	7

New Testament, book number

	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	4	4	3	4	3	10	8	10	9	9	4	8	9	10	8	6	6	6	6	6	10	9	7	5	8	8	2
Sentence	3	5	5	4	2	5	5	4	4	2	3	2	4	5	4	5	3	3	2	6	3	2	0	3	3	1	1
Paragraph	6	5	6	3	5	1	3	2	0	2	6	3	2	3	2	5	4	2	1	2	3	2	4	5	5	5	1

	Old Testament average	**New Testament average**	**Total average**
Word	9.23	6.74	8.21
Sentence	6.95	3.30	5.45
Paragraph	6.85	3.26	5.38

Table 5. Number of the same books among ten closest matches in the Greek and the Finnish 1933/1938 translations.

Old Testament, book number

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	7	6	4	7	3	6	5	7	4	6	9	7	6	6	4	6	6
Sentence	5	4	3	3	5	2	6	2	4	6	6	4	6	6	3	4	1
Paragraph	6	4	3	4	2	5	1	4	5	5	3	3	4	2	3	4	0

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	3	3	3	2	5	4	7	7	7	8	5	4	8	5	6	5	5	4	5	6	7	6
Sentence	6	4	4	5	5	5	6	4	3	6	4	1	4	4	1	1	2	4	3	4	6	4
Paragraph	4	3	4	2	2	4	3	3	4	1	4	1	0	0	2	3	3	1	1	2	1	3

New Testament, book number

	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	5	7	4	4	3	7	7	8	9	4	8	5	7	9	7	4	6	5	5	7	7	7	3	3	4	5	6
Sentence	5	7	5	7	7	3	5	3	4	2	3	3	1	1	2	5	2	3	6	5	4	5	4	1	2	4	5
Paragraph	5	5	1	4	4	0	2	3	3	2	2	4	1	1	2	2	4	3	4	1	1	0	1	2	0	2	2

	Old Testament average	**New Testament average**	**Total average**
Word	5.49	5.78	5.61
Sentence	4.00	3.85	3.94
Paragraph	2.79	2.26	2.58

Table 6. Number of the same books among ten closest matches in the Greek and the Finnish 1992 translations.

	Old Testament, book number																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	7	6	4	7	3	6	5	6	4	4	7	6	5	6	4	4	6
Sentence	6	4	6	6	6	4	7	0	8	8	7	7	6	7	2	4	4
Paragraph	2	2	4	2	3	3	2	2	5	4	5	3	5	4	3	3	2

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	4	2	3	2	3	4	5	6	7	6	5	2	5	4	6	5	5	5	5	4	7	5
Sentence	4	4	2	5	5	5	5	1	5	7	7	3	4	0	0	2	0	3	3	2	6	4
Paragraph	2	1	5	1	1	4	1	3	4	1	5	5	2	1	5	3	1	2	3	0	4	2

	New Testament, book number																										
	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	5	7	4	4	3	7	6	8	6	5	5	7	6	8	6	4	5	7	3	5	6	8	3	3	4	5	4
Sentence	4	6	4	7	6	4	6	3	4	4	5	2	2	2	4	3	4	1	4	5	3	3	3	1	0	7	6
Paragraph	4	3	4	2	4	1	2	2	1	0	3	3	0	1	1	2	0	2	3	1	2	0	4	1	2	0	1

	Old Testament average	**New Testament average**	**Total average**
Word	4.87	5.33	5.06
Sentence	4.33	3.81	4.12
Paragraph	2.82	1.81	2.41

Table 7. Number of the same books among ten closest matches in the Finnish 1933/1938 and the Finnish 1992 translations.

	Old Testament, book number																
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
Word	10	9	5	8	5	7	9	9	8	8	8	7	8	8	6	8	9
Sentence	7	6	5	6	7	5	8	3	5	6	5	5	7	7	4	5	6
Paragraph	1	3	2	3	4	3	3	5	5	4	3	4	5	5	4	2	1

	18	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	**35**	**36**	**37**	**38**	**39**
Word	5	8	8	6	6	7	7	6	6	7	7	7	5	7	9	9	8	7	9	7	7	8
Sentence	6	7	6	6	7	6	5	3	4	4	5	3	5	3	1	7	2	1	3	2	4	4
Paragraph	3	4	2	0	3	4	4	2	6	3	3	2	0	0	2	3	1	1	0	2	1	1

	New Testament, book number																										
	40	**41**	**42**	**43**	**44**	**45**	**46**	**47**	**48**	**49**	**50**	**51**	**52**	**53**	**54**	**55**	**56**	**57**	**58**	**59**	**60**	**61**	**62**	**63**	**64**	**65**	**66**
Word	9	9	10	6	8	7	7	8	7	8	7	7	8	8	8	5	6	6	6	7	8	6	7	8	6	5	4
Sentence	7	7	5	7	6	4	6	3	5	2	2	4	5	3	3	5	1	2	3	3	4	2	5	4	0	3	5
Paragraph	4	2	2	4	4	3	1	2	1	3	3	4	2	1	1	4	1	3	2	1	4	1	3	2	1	1	4

	Old Testament average	**New Testament average**	**Total average**
Word	7.38	7.07	7.26
Sentence	4.90	3.93	4.50
Paragraph	2.67	2.37	2.55

Table 8. Number of the books of the same class among five closest matches in the Greek translation.

Book	Word	Sentence	Paragraph	Book	Word	Sentence	Paragraph
18	1	2	2	**40**	3	3	0
19	0	2	2	**41**	2	3	0
20	1	1	0	**42**	3	3	1
21	0	1	1	**43**	3	3	2
22	1	1	0	**44**	2	0	0
Sum	3	7	5		13	12	3

Table 9. Number of the books of the same class among five closest matches in the Finnish 1933/1938 translation.

Book	Word	Sentence	Paragraph	Book	Word	Sentence	Paragraph
18	2	3	2	**40**	4	3	3
19	2	3	1	**41**	4	3	1
20	4	3	1	**42**	4	0	2
21	3	1	0	**43**	4	2	1
22	0	0	0	**44**	3	0	2
Sum	11	10	4		19	8	9

Table 10. Number of the books of the same class among five closest matches in the Finnish 1992 translation.

Book	Word	Sentence	Paragraph	Book	Word	Sentence	Paragraph
18	2	3	2	**40**	4	3	3
19	2	3	1	**41**	4	3	1
20	4	3	1	**42**	4	0	2
21	3	1	0	**43**	4	2	1
22	0	0	0	**44**	3	0	2
Sum	11	10	4		19	8	9

Statistical and Neural Approaches for Estimating Parameters of a Speckle Model Based on the Nakagami Distribution

Mark P. Wachowiak[1], Renata Smolíková[1,2], Mariofanna G. Milanova[1,3], and Adel S. Elmaghraby[1]

[1] Department of Computer Engineering and Computer Science
University of Louisville, Louisville, KY 40292, USA
{mpwach01, r0smol01, mgmila01, aselma01}@athena.louisville.edu
[2] Institute for Research and Applications of Fuzzy Modeling
University of Ostrava, Czech Republic
[3] ICSR, Bulgarian Academy of Science, Bulgaria

Abstract. The Nakagami distribution is a model for the backscattered ultrasound echo from tissues. The Nakagami shape parameter m has been shown to be useful in tissue characterization. Many approaches to estimating this parameter have been reported. In this paper, a maximum likelihood estimator (MLE) is derived, and a solution method is proposed. It is also shown that a neural network can be trained to recognize parameters directly from data. Accuracy and consistency of these new estimators are compared to those of the inverse normalized variance, Tolparev-Polyakov, and Lorenz estimators.

1 Introduction

Speckle noise, or simply speckle, is the greatest single factor that makes visual and computerized analysis of biomedical ultrasound (US) images difficult. However, the process that causes speckle has implications for interpretation and analysis of US signals. Speckle in ultrasound images is due to backscattering conditions, or the constructive-destructive interference of echoes returning to the US transducer after passing through and being reflected by tissue. Backscattering from scatterers (tissues, cells, or material that "scatters" acoustic waves) can be modeled as a random walk. Thus, this backscattered envelope of the echo is thought to follow specific statistical distributions. Various models for backscattering have been proposed, including the Rayleigh, Rician, K, homodyned K, generalized K, and Nakagami distributions [3], [7]. The parameters of these models can be used to characterize tissue regions. For example, regions can be classified as healthy or diseased based on the echo envelope statistics. In particular, the Nakagami distribution has been proposed as a general statistical model for ultrasonic backscattering because of its analytical simplicity (compared with the K and homodyned K distributions), and, when combined with phase analysis, its ability to model almost all scattering conditions [7].

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 196–205, 2001.

The Nakagami distribution is characterized by a shape parameter m and a scale parameter Ω, which is also the second moment of the distribution. The m parameter can provide information on scattering characteristics and scatterer density [7]. A large number of randomly spaced scatterers corresponds to the case of $m = 1$, and in fact the Nakagami model becomes a Rayleigh distribution. The case of $m > 1$ corresponds to random and periodic scatterers, and the Nakagami model approaches a Rician distribution. The special case of a small number of random scatterers can be modeled with $m < 0.5$. The ability of m to characterize scatterers decreases when $m > 2$. For these cases, all that can be said is that periodic and random scatterers are present.

This paper addresses estimation of the shape parameter m of the Nakagami distribution. Because m can be used in scatterer characterization, it can potentially provide important clinical and diagnostic information. In this study, a maximum likelihood estimator (MLE) is derived, along with a new solution method, and an approach based on neural learning is proposed. Experiments are performed on simulated data, and results are compared with three existing estimators.

2 Background

A random variable X is from the Nakagami distribution if its probability density function (pdf) is [6]:

$$f_X(x) = \frac{2m^m}{\Gamma(m)\Omega^m} x^{2m-1} \exp\left(\frac{-mx^2}{\Omega}\right), \quad x \geq 0,\, m \geq \frac{1}{2},\, \Omega > 0, \tag{1}$$

where Γ is the gamma function, m is the shape parameter and Ω is the scale parameter. Plots for different values of m with $\Omega = 1$ are shown in Fig. 1. Examples of Nakagami distributed data are shown in Fig. 2.

If a new random variable $Y = X^2$ is defined, then

$$f_Y(y) = \frac{m^m y^{m-1}}{\Gamma(m)\Omega^m} \exp\left(\frac{-my}{\Omega}\right), \quad y \geq 0,\, m > 0,\, \Omega > 0, \tag{2}$$

i.e., Y is Gamma distributed and the parameter m now takes values in $(0, \infty)$. This fact is used in generating Nakagami random variates, and for special cases of the Nakagami distribution when $m < \frac{1}{2}$ [7].

The k-th moment of the Nakagami pdf can be written as

$$E(X^k) = \frac{\Omega^{k/2}\Gamma(\frac{k}{2} + m)}{m^{k/2}\Gamma(m)}, \tag{3}$$

with $E(\cdot)$ denoting expectation. Hence, $E(X^2) = \Omega$, and

$$m = \frac{\Omega^2}{Var(X^2)} = \frac{1}{Var_N(X^2)}, \tag{4}$$

where $Var_N(X^2)$ denotes the normalized variance of X^2 [6]. The scale parameter Ω can be estimated from N samples as $\hat{\Omega} = \frac{1}{N}\sum_{i=1}^{N} x_i^2$.

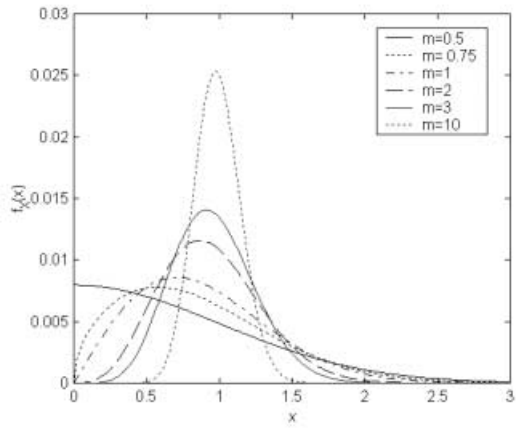
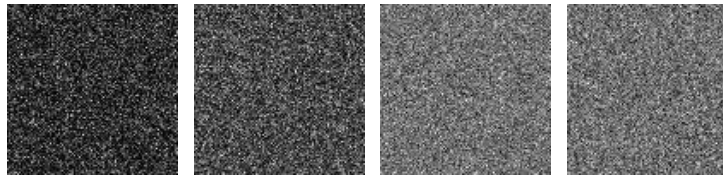


Fig. 1. Plots of the Nakagami pdf for six different values of m ($\Omega = 1$).



(a) $m = 0.5$ (b) $m = 1$ (c) $m = 10$ (d) $m = 20$

Fig. 2. Nakagami distributed data for various m values and $\Omega = 1$.

3 Parameter Estimation Methods

Three common estimators for m are now described [1]. Let $\hat{\mu}_k = \frac{1}{N}\sum_{i=1}^{N} x_i^k$ denote the estimate of the k-th moment of a sample $\{x_i | i = 1, \ldots, N\}$, which is a set of realizations of N statistically independent random variables $X_i, i = 1, \ldots, N$. A general moment based approach can be used to obtain an estimate $\hat{m}$ [1]:

$$\frac{\Gamma(\hat{m}_k + k/2)}{\Gamma(\hat{m}_k)\hat{m}_k^{k/2}} = \frac{\hat{\mu}_k}{\hat{\mu}_2^{k/2}}, \qquad k = 1, 2, \ldots \tag{5}$$

When $k = 4$, the inverse normalized variance (INV) estimator, $\hat{m}_{\mathrm{INV}}$, is obtained from Eq. 5 [1]:

$$\hat{m}_{\mathrm{INV}} = \frac{\hat{\mu}_2^2}{\hat{\mu}_4 - \hat{\mu}_2^2}. \tag{6}$$

The second method, the Tolparev-Polyakov (TP) estimator $\hat{m}_{\mathrm{TP}}$, is expressed as [1]:

$$\hat{m}_{\mathrm{TP}} = \frac{1 + \sqrt{1 + (4/3)\ln(\hat{\mu}_2/B)}}{4\ln(\hat{\mu}_2/B)}, \tag{7}$$

where $B = (\prod_{i=1}^{N} x_i^2)^{1/N}$. A third estimator, the Lorenz (L) estimator $\hat{m}_{\mathrm{L}}$, is given by [1]:

$$\hat{m}_{\mathrm{L}} = \frac{4.4}{\sqrt{\hat{\mu}_2^{dB} - (\hat{\mu}_1^{dB})^2}} + \frac{17.4}{[\hat{\mu}_2^{dB} - (\hat{\mu}_1^{dB})^2)]^{1.29}}. \tag{8}$$

In Eq. 8, $\hat{\mu}_k^{dB} = N^{-1} \sum_{i=1}^{N} (20 \log x_i)^k$.

4 Maximum Likelihood Method

In maximum likelihood estimation (MLE), an estimate of an unknown parameter is a value in the parameter space that corresponds to the largest "likelihood" for the observed data. The likelihood is expressed by a likelihood function. Let X_i denote random variables that are identically and independently distributed according to Eq. 1. For the Nakagami distribution, the likelihood function is

$$L(m, \Omega) = \prod_{i=1}^{N} f_{X_i}(x_i) = \left(\frac{2m^m}{\Gamma(m)\Omega^m}\right)^N \left(\prod_{i=1}^{N} x_i^{2m-1}\right) \exp\left(-\frac{m}{\Omega}\sum_{i=1}^{N} x_i^2\right), \tag{9}$$

and the log-likelihood function is expressed as

$$\ln L(m, \Omega) = N \ln\left(\frac{2m^m}{\Gamma(m)\Omega^m}\right) + (2m-1)\sum_{i=1}^{N} \ln x_i - \frac{m}{\Omega}\sum_{i=1}^{N} x_i^2. \tag{10}$$

The partial derivatives of the log-likelihood function are given by

$$\frac{\partial \ln L(m, \Omega)}{\partial \Omega} = -\frac{mN}{\Omega} + \frac{m}{\Omega^2}\sum_{i=1}^{N} x_i^2, \tag{11}$$

$$\frac{\partial \ln L(m, \Omega)}{\partial m} = N \ln\frac{m}{\Omega} + N - N\psi(m) + 2\sum_{i=1}^{N} \ln x_i - \frac{1}{\Omega}\sum_{i=1}^{N} x_i^2. \tag{12}$$

Here, $\psi(x) = \frac{d}{dx} \ln \Gamma(x)$ is the digamma function. An estimate for Ω, $\hat{\Omega} = \frac{1}{N}\sum_{i=1}^{N} x_i^2$, is obtained by equating Eq. 11 to zero. Substituting $\hat{\Omega}$ into Eq. 12 and equating to zero gives

$$\ln m - \psi(m) = \ln\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \frac{2}{N}\sum_{i=1}^{N} \ln x_i. \tag{13}$$

The L.H.S. of Eq. 13 is a transcendental function that cannot be solved analytically for m. However, m can be computed numerically if the inverse of Eq. 13 exists. To verify the existence of the inverse, the derivative of the L.H.S. of Eq. 13 is computed:

$$\frac{\mathrm{d}(\ln m - \psi(m))}{\mathrm{d}m} = \frac{1}{m} - \psi^{(1)}(m), \tag{14}$$

where $\psi^{(1)}(m)$ is the first derivative of $\psi(m)$. It has been shown that [2]

$$(-1)^{k+1}\psi^{(k)}(x) = k! \sum_{i=0}^{\infty} \frac{1}{(x+i)^{(k+1)}} > \frac{(k-1)!}{x^k}, \tag{15}$$

with $x \in \mathcal{R}^+$, $k \geq 1$ and $\psi^{(k)}(x)$ denoting the k-th derivative of $\psi(x)$. Substituting $k = 1$, $x = m$, and simplifying gives

$$\frac{1}{m} < \psi^{(1)}(m) \Rightarrow \frac{1}{m} - \psi^{(1)}(m) < 0. \tag{16}$$

Thus, as its derivative is negative for all $m > 0$, $g(m) = \ln(m) - \psi(m)$ is a strictly decreasing function, and therefore its inverse exists. The MLE estimate of m, $\hat{m}_{\mathrm{MLE}}$, is then written as

$$\hat{m}_{\mathrm{MLE}} = g^{-1}\left(\ln\left(\frac{1}{N}\sum_{i=1}^{N} x_i^2\right) - \frac{2}{N}\sum_{i=1}^{N}\ln x_i\right). \tag{17}$$

As will be shown, Eq. 17 allows $\hat{m}_{\mathrm{MLE}}$ to be computed with simple numerical techniques such as spline or even linear interpolation.

5 Parameter Estimation with Neural Networks

An important aspect of a statistical distribution is its shape, which is dependent on its parameter, as can be seen from Fig. 1. Thus, estimation can be formulated as a pattern recognition problem. A neural network is trained to estimate the parameters from the histogram of a data set. Neural networks have proven to be very powerful in pattern recognition and in parameter estimation [5], and previous work has proposed neural techniques for US speckle characterization [8], [9], [10]. Many architectures, such as radial basis function networks and generalized regression networks, could be used for this task. However, a simple feedforward architecture was selected because the weights can then be used in a matrix multiplication formulation; that is, the trained network can perform faster than kernel-based networks. The resulting network can also easily be implemented in hardware or on special-purpose computers. Furthermore, such networks have proven very successful in pattern recognition and in function interpolation. The network contains 30 input units (the data histogram contained 30 evenly-spaced bins ranging in value from 0 to 10), 5 hidden neurons, and 1 output neuron representing the m parameter. The network was trained with simulated Nakagami data with $m \in [0.1, 40]$, using backpropagation learning.

6 Methods

Five-hundred random values of m, different from those used for training, were generated in the range [0.1, 40]. Nakagami distributed random variates with these parameters are generated as $X = \sqrt{Y}$, where Y is gamma distributed with parameter m (see Eq. 2) [1]. Using this formulation, it is possible to generate special cases of the Nakagami distribution where m lies between 0 and 0.5. This distribution can be called the Nakagami-Gamma distribution to distinguish it from the strict Nakagami model wherein $m \geq 0.5$ [7]. The scale parameter Ω is set to unity in all experiments. In actual parameter estimation, the data can be normalized by $\sqrt{\hat{\mu}_2}$ to obtain $\hat{\Omega} = 1$.

The INV, TP, L, MLE, and artificial neural network (ANN) estimators were applied to the simulated data of sizes $N = 100$ (10×10 pixels), 1000, 2500, and 10000 (100×100 pixels). One hundred trials were performed for all five estimators for each of the 500 parameters. The mean and standard deviation of the estimate $\hat{m}$ were computed for the five methods and the four N. For the MLE method, a consequence of Eq. 17 is that $\hat{m}_{\mathrm{MLE}}$ can be computed with simple numerical interpolation techniques. For instance, $g(m) = \ln m - \psi(m)$ can be numerically computed for values of m in a finely-spaced grid in the range [0.1, 10]. An interpolation algorithm can then be applied that treats $g(m)$, estimated from the L.H.S. of Eq. 13, as the independent variable. The interpolated value is $\hat{m}_{\mathrm{MLE}}$. In the current study, cubic spline interpolation was used [4].

7 Results

Experimental results are shown in Figs. 3 and 4, and in Table 1. The figures show the mean of $\hat{m}$ for the 500 m values and the standard deviation of these estimates versus the true m value. Table 1 shows the root mean squared error (E_{RMS}) values for $m \in [0.1, 40]$, and for small m ($m \in [0.1, 5]$). The table also shows the mean of the standard deviation over the 500 $\hat{m}$ values computed with the various methods.

From the experimental data, INV, TP, and ANN have the best E_{RMS} values for all m. For all methods, E_{RMS} increases with smaller sample sizes, especially for $N = 100$ and $m > 5$. The L estimator shows a positive bias with increasing m, as also reported in [1]. However, unlike the results in [1], the current study does not show a large performance advantage of the INV estimator as compared to TP.

The ANN estimator shows more consistency (lower standard deviation) than the other methods over all m, while all methods were generally consistent for small m. In fact, the ANN estimator appears to perform better, both in E_{RMS} and consistency measures, for $m >> 1$. For small m, there is no clear top performer.

For $m \in [0.1, 5]$, TP and MLE provide slightly more consistent estimates for $N = 100$ than the other methods. For larger m ($m \in [20, 30]$), the neural estimator was the best performer for $N = 100$. The MLE estimator performed about as well as INV, TP, and ANN for small m.

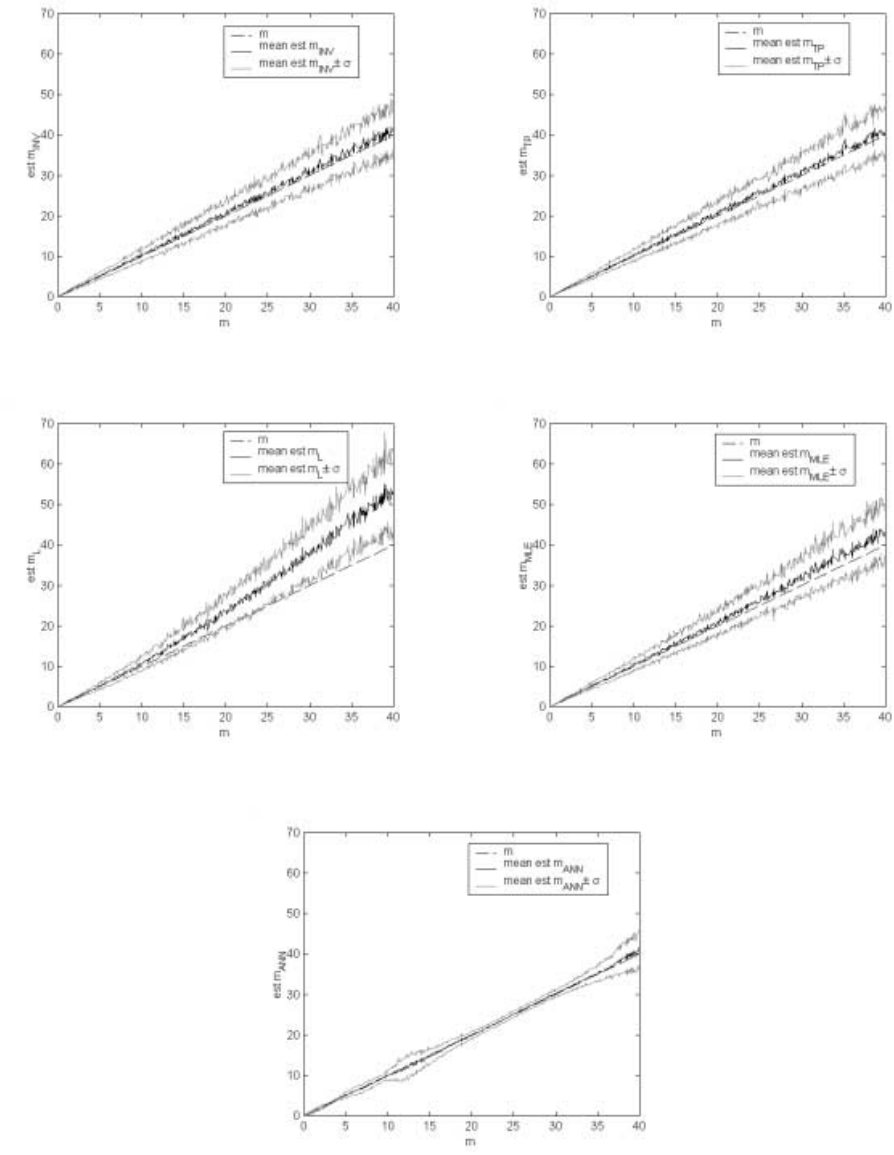


Fig. 3. Plots for five estimators, $N = 100$. - - denotes m, — denotes mean $\hat{m}$, and $\cdots$ denotes mean $\hat{m} \pm \sigma$.

Considering computational complexity, the INV, TP, and L estimators are computed with relatively simple closed-form expressions, although moment functions must be computed. The MLE approach is slightly more complex as, in addition to moment functions, an interpolation operation must be performed. However, preliminary experiments show that linear interpolation, which is much

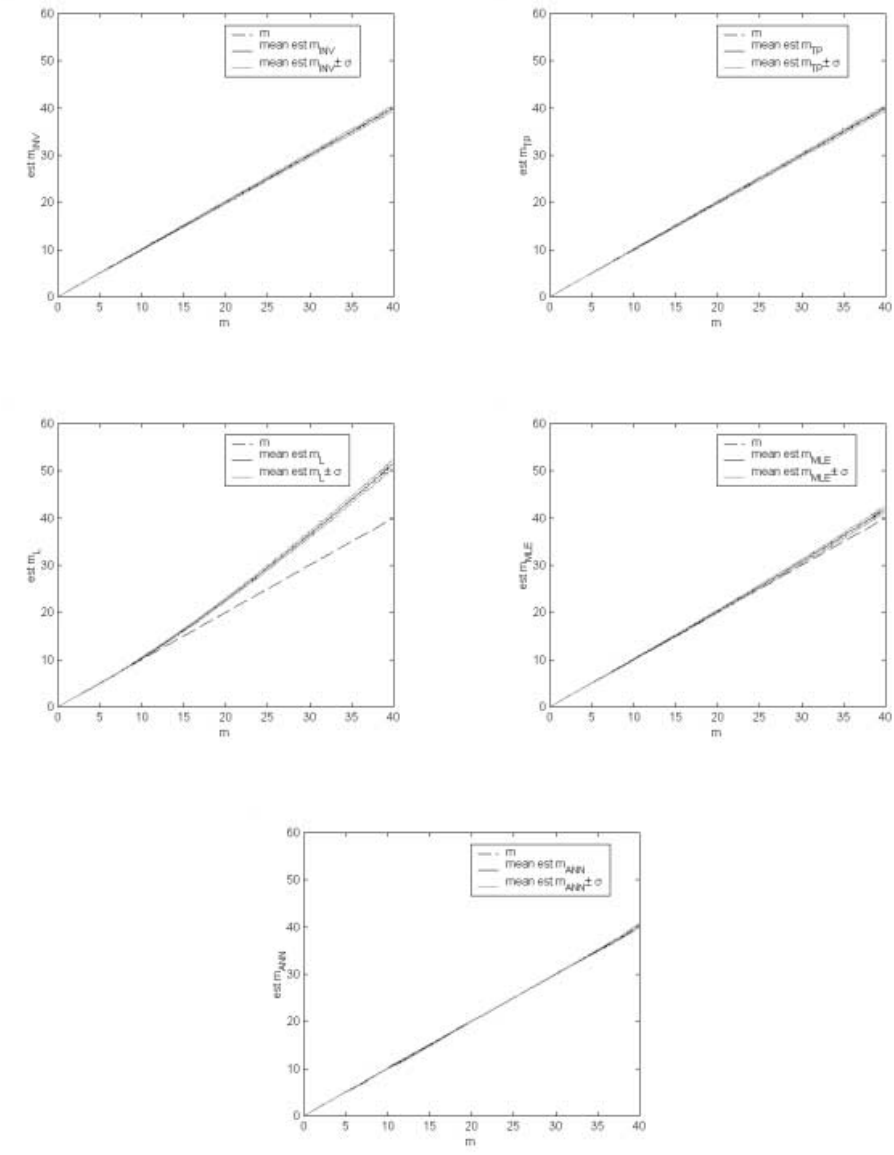


Fig. 4. Plots for five estimators, $N = 10000$. - - denotes m, — denotes mean $\hat{m}$, and $\cdots$ denotes mean $\hat{m} \pm \sigma$.

less complex than spline interpolation, also provides accurate m estimates. The ANN approach only requires normalization and binning the data. No moment functions need to be computed. As stated earlier, the ANN estimator can be implemented with fast matrix operators, or even in hardware.

Table 1. E_{RMS} and mean standard deviation for five different estimators, $m \in [0.1, 40]$, $m \in [0.1, 5]$.

		$m \in [0.1, 40]$					$m \in [0.1, 5]$				
	N	INV	TP	L	MLE	ANN	INV	TP	L	MLE	ANN
E_{RMS}	100	0.91	0.86	6.39	1.65	0.35	0.14	0.09	0.10	0.11	0.08
	1000	0.16	0.17	5.32	0.84	0.10	0.02	0.02	0.07	0.02	0.02
	2500	0.09	0.10	5.25	0.78	0.08	0.01	0.01	0.07	0.01	0.02
	10000	0.05	0.05	5.21	0.74	0.07	0.01	0.01	0.07	0.01	0.02
mean	100	3.05	2.97	4.25	3.28	1.41	0.44	0.36	0.39	0.38	0.40
std.	1000	0.92	0.91	1.26	0.97	0.43	0.14	0.11	0.12	0.11	0.12
dev.	2500	0.58	0.57	0.80	0.61	0.28	0.09	0.07	0.07	0.07	0.07
	10000	0.30	0.29	0.40	0.30	0.14	0.04	0.03	0.04	0.03	0.04

8 Conclusions

An MLE formulation, along with a new solution approach, and novel estimation technique based on neural learning is presented in this paper. These estimators are compared with three existing methods. The neural network consistently gives the best estimates for large values of m. For small m, the ANN and MLE estimators also perform very well, as do existing methods, especially TP and INV. It has been demonstrated that the MLE and ANN estimators can be used for determining m in ultrasound applications with a high degree of confidence. The superior performance of the ANN estimator for high m values is due to (1) its ability to generalize, as estimates were very good for m values that were not used in training, and (2) its ability to recognize small differences in similar data (see Figs. 2c and 2d). However, if time complexity is the primary concern, the INV or TP estimators are slightly preferable.

This study shows that MLE and neural estimators can complement existing techniques. The results suggest that, for US applications, neural networks may be trained with an even more limited range of m to further increase estimation accuracy for US parameter estimation, since, in these cases, m will rarely be larger than 2. The results also demonstrate the efficacy, in general, of neural approaches for parameter estimation from data. Because of the importance of accurate tissue characterization in speckled US images, parameter estimation, including neural approaches, merit further research.

References

1. Abdi, A., Kaveh, M.: Performance Comparison of Three Different Estimators for the Nakagami m Parameter Using Monte Carlo Simulation. IEEE Communications Letters **4** (2000) 119–121
2. Alzer, H.: On Some Inequalities for the Gamma and Psi Functions. Math. Comp. **66** (1997) 373–389

3. Clifford, L., Fitzgerald, P., James, D.: Non-Rayleigh First-Order Statistics of Ultrasonic Backscatter from Normal Myocardum. Ultrasound in Med. and Biol. **19** (1993) 487–495
4. Iskander, D. R., Zoubir, A. M., Boashash, B.: A Method for Estimating the Parameters of the K Distribution. IEEE Trans. Sig. Proc. **47** (1991) 1147–1151
5. Liu, M. C., Kuo, W., Sastri, T.: An Exploratory Study of a Neural Network Approach for Reliability Data. Analysis Quality and Reliability Eng. Intl. **11** (1995) 107–112
6. Nakagami, M.: The m-distribution: A general formula of intensity distribution of rapid fading. in Statistical Methods in Radio Wave Propagation W. C. Hoffman, Ed. New York: Pergamon (1960) 3–36
7. Shankar, P. M.: A General Statistical Model for Ultrasonic Backscattering from Tissues. IEEE Trans. on Ultrasonics, Ferroelectrics and Frequency Control **47** (2000) 727–736
8. Smolíková, R., Wachowiak, M. P., Elmaghraby, A. S., Zurada, J. M.: A Neuro-Statistical Approach to Ultrasound Speckle Modeling. Proc. ISCA 13^{th} Intl. Conf., Honolulu, HI (2000) 94-97
9. Wachowiak, M. P., Smolíková, R., Zurada, J. M., Elmaghraby, A. S.: A Neural Approach to Speckle Noise Modeling. Intelligent Engineering System Through Artificial Neural Networks: Smart Engineering System Design: Neural Networks, Fuzzy Logic, Evolutionary Programming, Data Mining and Complex Systems **10** ASME Press, New York (2000) 837–842
10. Wachowiak, M. P., Smolíková, R., Elmaghraby, A. S., Zurada, J. M.: Classification and Estimation of Ultrasound Speckle Noise With Neural Networks. Proc. on IEEE International Symposium on Bio-Informatics and Biomedical Engineering. Arlington, Virginia (2000) 245–252

How to Automate Neural Net Based Learning

Roland Linder and Siegfried J. Pöppl

Institute for Medical Informatics, Medical University of Luebeck
Ratzeburger Allee 160, D-23538 Luebeck

Abstract. Although neural networks have many appealing properties, yet there is neither a systematic way how to set up the topology of a neural network nor how to determine its various learning parameters. Thus an expert is needed for fine tuning. If neural network applications should not be realisable only for publications but in real life, fine tuning must become unnecessary. In the present paper an approach is demonstrated fulfilling this demand. Moreover referring to six medical classification and approximation problems of the PROBEN1 benchmark collection this approach will be shown even to outperform fine tuned networks.

1 Introduction

For applications in medicine multilayer perceptrons (MLP) trained by the backpropagation algorithm [1] are most popular. Despite the general success of back-propagation in learning neural networks [2] several deficiencies are still needed to be solved. Learning can be trapped into local minima, the training process does converge slowly, and there are difficulties in explaining the network's response. Since 1996 when Rumelhart et al. had introduced backpropagation, various coping strategies have been published [3]. Some of these strategies will be presented in this paper, existing ones as well as newly developed approaches. But the most apparent disadvantage is that the convergence behaviour depends very much on the choice of the network topology and diverse parameters in the algorithm such as the learning rate and the momentum. Therefore the presence of an expert seems to be absolutely necessary. The need for fine tuning may be the greatest obstacle for a wide-spread use of neural network techniques in medicine.

Attempts in designing at least the network structure automatically have been undertaken by various constructive algorithms [4], that can be roughly divided into dynamic node creation (DNC) [5], and cascade correlation (CC) [6]. While the DNC-like algorithms are computationally expensive, the CC-like algorithms have problems in always finding a good solution [7, 8] and may be unsuitable for rule extraction. Assuming the future availability of hard-coded neural networks the automatic training of a fixed network architecture remains highly desirable.

Therefore we will present an approach that mainly relies on an expanded version of a multi-neural-network architecture by Anand et al. [9] in connection with *adaptive propagation* [10, 11], an improvement of the backpropagation algorithm. This network

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 206–216, 2001.

can be trained without any fine tuning. Its performance will be demonstrated by solving five medical multiclass classification problems and one medical approximation problem. These problems are part of the established PROBEN1 benchmark collection from Prechelt [12]. Moreover we will prove the usefulness of an ensemble of multi-neural-networks.

The organisation of this paper is as follows. Section 2 and 3 describe strategies used for our approach, whereas in section 2 we will concentrate on strategies for improving the generalisation performance and section 3 describes how to accelerate learning. Section 4 gives a short description how the algorithm was implemented, and an introduction to the benchmarks used. Simulation results are given in section 5 and finally conclusions are drawn in section 6.

2 Strategies for Improving the Generalisation Performance

For purposes of more clarity, strategies for improving the generalisation performance will be separately listed from those accelerating the convergence speed. Naturally overlapping can not be avoided.

2.1 Multi-neural-Network Architecture

An approach published by Anand et al. [9] is to use a modular network architecture for multiclass classification problems. In this architecture each module is a single-output network which determines whether a pattern belongs to a particular class, thereby reducing a *k*-class problem to a set of *k* two-class problems. A module for class C_k is trained to distinguish between patterns belonging to C_k and its complement $\overline{C}_k$. In general $\overline{C}_k$ will have many more patterns than C_k. Therefore the output errors must be weighted in order to equalise the importance given to each class. When training not approximately the same number of patterns per class (as Anand did), the a-priori probabilities of each class must be taken into account by feeding a further MLP with the outputs of the modules. This additional MLP comprises *k* input and *k* output neurons and means a modification of Anand's approach. The modular approach has the following advantages:

1. It is easier to learn a set of simple functions separately than to learn a complex function which is a combination of the simple functions. In some cases training non-modular networks is unsuccessful even when indefinitely long training periods are permitted, whereas modular networks do converge successfully.
2. In a nonmodular network conflicting signals from different output nodes retard learning. Modular learning is likely to be more efficient since weight modification is guided by only one output node. Moreover the modules can be trained independently and in parallel. Software simulations of modular neural networks can therefore utilise massively parallel computing systems much more effectively than nonmodular networks.

3. Explaining the results of a modular network will be easier than for a nonmodular network, since the relation between input neurons and the output neuron(s) is easier to establish (by examining the connection weights) in each module.

Moreover the additional MLP may improve the generalisation performance in the way of a cascaded architecture in the sense of Qian and Sejnowski [13].

2.2 Refining the Target Output

Usually target outputs are *1-out-of-C* coded where an output with *C* unordered categories is converted into *C* Boolean outputs, each of them is *one* only for a certain category, otherwise *null*. Indeed most often the assessment of null does not reflect the truth because very rarely there are sharp boundaries between adjacent classes. Due to the impossibility to fit the real outputs exactly to the desired nulls, the mean squared error (MSE) can not converge to null during training the network. Even when 0.1 instead of 0.0 can be achieved on average, this means a relevant contribution to the MSE (because most target outputs are set at null). As pre-tests have demonstrated this contribution to the MSE hampers the network in concentrating on the still misclassified patterns. A remedy can be found by refining the target output after a predefined number of learning epochs: All target values equal to null with a difference of not more than e.g. 0.5 from the real outputs will be set at the corresponding real output values for further training. We suggest to do so for the additional MLP (see Chapter 2.1). Solving a multiclass classification problem it is often more important to assign a certain pattern to the correct class than obtaining the minimum mean squared error.

2.3 Early Stopping

During the learning phase the error on the training set decreases more or less steadily, while the error on unseen patterns starts at some point - usually in the later stages of learning - to get worse again. Before reaching this point the network learns the general characteristics of the classes, afterwards it takes advantage of some idiosyncrasies in the training data worsening the generalisation performance. Several theoretical works have been done on the optimal stopping time [14, 15]. One approach to avoid this so-called overfitting is to estimate the generalisation ability during training (with an extra validation set removing some patterns from the training data) and to stop when it begins to decrease. This widely used technique is called *early stopping* and has been reported to be superior to other regularisation methods in many cases, e.g. in Finnoff et al. [16]. However the real situation is somewhat more complex. Real generalisation curves almost always have more than one local minimum. Therefore Prechelt distinguishes 14 different automatic stopping criteria [17]. For the present approach learning was stopped after a predefined number of epochs (1000) and the test set performance was then computed for that state of the network which had the minimum validation set error during the training process.

2.4 Avoiding Weight Sets from Phases of Oscillation

Sometimes - usually when the learning rate is chosen too high - the training process and also the generalisation curve become oscillating. In order to avoid testing with a state of the network when the minimum validation set error was thus achieved "at random", we demand a minimum number of epochs that show a continuous improvement of the validation set error directly before the state in question. Ten epochs are a good choice.

2.5 Network Ensemble

Most often in medicine there is only a small amount of data available. In order not to waste valuable data we suggest to make use of a network ensemble consisting of five multi-neural-networks as described above. So all training data have to be divided into five equally sized sets A to E. The first multi-neural-network will be trained by set A, B, C, and D. Set E will serve as a validation set. Training the second multi-neural-network, the training set consists of set A, B, C, and E, set D will be the validation set and so on. To get one common result for each class, the output activities of the five multi-neural-networks can be averaged. It is appropriate to neglect the minimum and maximum output value (in case that one of the multi-neural-networks fails). The mean value will be calculated only averaging the three remaining output activities.

As a welcome side-effect network ensembles are naturally more robust against unsuccessful runs than single networks.

2.6 Adaptive Propagation

As quick alternatives to slow standard backpropagation there have been proposed numerous algorithms like RPROP by Riedmiller and Braun [18], which is among the fastest gradient step size adaptation methods for batch backpropagation learning, or the rather little-known but very fast Vario-Eta by Finnoff et al. [19]. We developed an algorithm called *adaptive propagation* (APROP), useful not only to accelerate the convergence speed but also for improving the generalisation performance. APROP is based on the idea that within a society single individuals as well as the entire population benefit most when the process concentrates especially on successful individuals. APROP prefers adapting those weights that lead to successful neurons. To calculate the success $S_{l,n}$ of a neuron n in a layer l, its squared errors δ of the current epoch have to be added. The reciprocal value of the squared root of this sum makes the success $S_{l,n}$. p designates the number of training patterns.

$$S_{l,n} = \frac{1}{\sqrt{\sum_{i=1}^{p} \delta_{l,n,i}^2}} \qquad (1)$$

The success of a neuron is therefore defined by the amount and the distribution of its errors. For each neuron a local neuron-specific learning rate $\sigma_{l,n}$ has to be calculated. In principle that is the global learning rate σ times the success of the neuron. Therefore the adjustment of each weight depends on the local learning rate $\sigma_{l,n}$ of the particular neuron the connection is leading to. For a detailed description and benchmarking please refer to [10, 11].

2.7 Modifying the Error Function

In medicine there are frequently different prior class probabilities. In order to take also small classes into account sufficiently, we suggest a modified error function (squared δ-errors of the output neurons with keeping their sign):

$$\delta_{output\ \ layer} = f'\left(activation\ \ function\right) \cdot sign\left(t-o\right) \cdot \left(t-o\right)^2 \qquad (2)$$

$\delta_{output\ layer}$ designates the error δ of an output neuron, t denotes the target output, and o the real output. The effectiveness of this modification was examined in [20].

2.8 Squaring the Derivation

The main reason for getting trapped in a local minimum is due to the derivative of the activation function, i.e. $o \cdot \left(1-o\right)$ for the standard logistic sigmoid function, and $\left(1-o^2\right)$ for the hyperbolic tangent. When the actual output is approaching to either extreme values, the derivative of the activation function will be vanished, and the back propagated error signal will become very small. Thus the output can be maximally wrong without producing a large error signal. Then the algorithm may be trapped into a local minimum. Consequently the learning process and weight adjustment of the algorithm will be very slow or even suppressed. In accordance with the *generalised back-propagation algorithm* by Ng et al. [21] we propose to square the derivation so as to improve the convergence of the learning by preventing the error signal to be dropped to a very small value.

2.9 Varying the Learning Rate

Choosing an appropriate learning rate is one of the most important aspects not only with regard to convergence speed but also for obtaining a good generalisation performance. After a few years of enthusiasm about numerous kinds of line search procedures there is some disillusionment: Even when such methods are undoubtedly very quick, the sequence of weight vectors may converge to a bad local minimum, because the line search algorithms moves towards the bottom of whatever valley they reach. The reason is that “escaping” a local minimum requires an increase in the

overall error function, which is excluded by the line search procedures. For backpropagation or *adaptive propagation* - each with momentum - a constant learning rate regardless of the shape of the error surface is used, which may lead to a "jump" over a local minimum. In order to set this constant at the most suitable value, the learning rate can be automatically varied for different runs - e.g. ten times - using an "intelligent" strategy finding the optimum value. The MSE of the validation set may serve as a criterion for finding out the best learning rate. Using ensembles of multi-neural-networks the learning rate can be optimised for each network within the ensemble as well as for each module within the multi-neural-network. However as a pre-condition the validation set must be large and representative. Otherwise the neural network will be biased by the validation set, since the selected network has indirectly adapted itself to the validation set. As a rule of thumb we suggest only to vary the learning rate if the number of validation records exceeds 1000.

3 Strategies for Accelerating the Convergence Speed

In order to accelerate the convergence speed, numerous methods have been proposed [22]. For the universal approach presented here we suggest to consider the following aspects.

3.1 Oversizing the Network Architecture

It is well known that a fast convergence speed can be achieved by oversizing the network structure. However most researchers follow the spirit of Occam's razor [23] and choose the smallest admissible size that will provide a solution, because in their opinion the simplest architecture is the best for generalisation. Notwithstanding several neural net empiricists have published papers showing that surprisingly good generalisation can in fact be achieved with oversized multilayer networks [24, 25, 26]. Already in 1993 Caruana [27] reported that large networks rarely did worse than small networks on the problems he investigated. Caruana suggested that „backprop ignores excess parameters". Also Rumelhart wrote: "Adding a few more connections creates extra dimensions in weight-space and these dimensions provide paths around the barriers that create poor local minima in the lower dimensional subspaces." [1]. From our experience oversizing the network architecture leads to a dramatic increase of convergence speed as well as to an improved generalisation performance (assuming early stopping, see Chapter 2.3). In the following a network will be chosen that comprises only one hidden layer (facilitates rule extraction later on) but 100 hidden neurons.

On the side, (1) oversized networks make the suitability of the weight initialisation more unaffected by random, and (2) guarantee a sufficient approximation capacity when learning different classification tasks of varying complexity using always the same number of hidden neurons (as we do, see Chapter 4).

3.2 APROP for Speeding Up

Due to the above described basic idea of APROP, this algorithm is ideally suited for oversized networks. When compared to algorithms like RPROP, Vario-Eta, or Quasi-Newton techniques [10, 11] APROP could save up to three-quarters of learning time needed by the other algorithms.

3.3 Stop Oscillating and Unsuccessful Learning

In order to save learning time, training should be stopped after a predefined number of oscillating epochs in the generalisation curve - e.g. ten epochs - or after a maximum number of epochs leading to no further decrease of the validation set error, say 100 epochs at a total amount of 1000 learning epochs.

4 Implementation and Benchmarks

The proposed algorithm was implemented as a prototype using the programming language MS Visual C++ 6.0 SP 4 and a Pentium III-1000 DP. Because C++ is not very comfortable in programming the graphical user interface, MS Visual Basic was used for creating the GUI, calling C++ DLLs that contain the actual neural network code. The compiler settings were optimised for speed, the multithreaded code was optimised by an extensive use of pointers, small dimensioned arrays, avoidance of if-statements or consecutive instructions that hamper pipelining. Also an approximation of the exponential function proposed by Schraudolph [28] has been proved to speed up calculation.

The algorithm was realised as already suggested above. All experience was gained by evaluating signals from an electronic nose [20]. None of our suggestions was influenced by the benchmarks presented here. Otherwise our results might have been distorted. As the only pre-processing data were z-transformed. Carefully all weights were initialised randomly within an interval of [-0.01, +0.01]. As activation function for the hidden neurons a hyperbolic tangent was employed. Its advantage over the standard logistic sigmoid function is the symmetry of its outputs with respect to null [29]. The standard logistic sigmoid function was used for the output neurons. There were no shortcut connections in order not to disturb the building up of an internal hierarchy. Learning was performed as batch learning. After the second epoch [30], a momentum term was utilised with a momentum factor μ set at 0.9. The global learning rate $\hat{\eta}$ was set at 0.1, c was set at 5, ε_n was initialised randomly within [0, 1], and a_n was initialised randomly within [-2, +2]. For a detailed description of these APROP-specific constants please refer to [10, 11]. As pre-tests demonstrated there was no need to start several runs from different weight initialisation (see Chapter 3.1). Thus only two runs where done per benchmark: the first one used all data excluding the test data and employed a network ensemble. The second one used exactly the same learning and validation set as demanded for the benchmark.

The performance was tested by solving five multiclass classification problems and one medical approximation problem of the standardised PROBEN1 benchmark collection from Prechelt [12]. These six of thirteen benchmarks are those that refer to real medical classification tasks:

cancer: Diagnosis of breast cancer. The aim is to classify a tumour either benign or malignant based on cell descriptions gathered by microscopic examination.

diabetes: Diagnosis of diabetes of Pima Indians. Based on personal data and the results of medical examinations it has to be decided whether a Pima Indian individual is diabetes positive or not.

gene: Detection of intron / exon boundaries (slice junctions) in nucleotide sequences. From a window of 60 nucleotides one has to decide whether the middle is either an intron / exon boundary (a donor), or an exon / intron boundary (an acceptor), or none of these.

heartc: Prediction of heart disease. The aim is to decide whether at least one of four major vessels is reduced in diameter by more than 50%. The binary decision is made based on personal data, subjective pain descriptions, and results of various medical examinations.

thyroid: Diagnosis of thyroid hyper- or hypofunction. Based on patient query data and patient examination data, the task is to decide whether the patient's thyroid has overfunction, normal function, or underfunction.

heartac: Differently from *heartc* the benchmark *heartac* uses a single continuous output that represents the number of vessels that are reduced.

Table 1 gives an overview of the number of inputs, outputs, and examples available. The data sets contain binary inputs as well as continuous ones. For each data set the total amount of examples was divided into three partitions: a learning set (50%), a validation set (25%), and a test set (25%). For each benchmark PROBEN1 contains three different permutations which differ only in the ordering of examples, e.g. *cancer1*, *cancer2*, and *cancer3*.

Table 1. Properties of the benchmarks used

Benchmark	*cancer*	*diabetes*	*gene*	*heartc*	*thyroid*
Inputs	9	8	120	35	21
Outputs	2	2	3	2	3
Examples	699	768	3175	303	7200

As fine tuning for each problem, Prechelt used twelve different MLP topologies (comprising 2, 4, 8, 16, 24, 32, 2+2, 4+2, 4+4, 8+4, 8+8, and 16+8 hidden nodes), experimented with linear output nodes and those using the sigmoid activation function, and he proved shortcut connections to be effective or not. For each benchmark Prechelt chose the architecture achieving the smallest validation set error. For a detailed description of architecture and learning parameters please refer to [12]. As classification method *winner-takes-all* was used, i.e. the output with the highest activation designates the class. For the approximation tasks Prechelt defined a *squared error percentage* that is similar to the MSE.

5 Results

Using the proposed network ensemble the percentages of misclassifications and the squared error percentages were significantly smaller than those of the manually designed MLP by Prechelt (Wilcoxon signed ranks test, p=0.026). Without ensemble and thus using the same validation set as Prechelt did, our results were also significantly better than the results by Prechelt's fine tuned MLP (same significance value p=0.026), see Table 2.

Table 2. Comparison of the percentages of misclassification or the squared error percentages

Benchmark	Tuned by Prechelt	With ensemble	Without ensemble
cancer1	1.38	2.30	2.87
cancer2	4.77	4.02	4.02
cancer3	3.70	4.02	4.02
diabetes1	24.10	23.44	23.44
diabetes2	26.42	22.92	24.48
diabetes3	22.59	21.53	22.40
gene1	16.67	11.48	11.48
gene2	18.41	8.45	8.95
gene3	21.82	10.34	11.22
heartc1	20.82	17.33	17.33
heartc2	5.13	6.67	4.00
heartc3	15.40	12.00	14.67
thyroid1	2.38	1.83	6.00
thyroid2	1.86	1.67	1.67
thyroid3	2.09	2.39	2.28
heartac1	2.47	2.29	2.26
heartac2	4.41	3.04	3.06
heartac3	5.37	3.78	4.05

The learning time varied corresponding to the benchmark calculated, e.g. learning *cancer1* with a network ensemble took 2:15 minutes.

6 Conclusions

The aim was to develop an universal approach that makes fine tuning unnecessary. Contrary to expectation this approach could be shown not only to achieve the same generalisation performance as Prechelt did when manually designing his MLP, but even to outperform his results in a statistically significant way. Due to the small number of output neurons needed for the benchmarks used above, the multi-neural-network approach might be even more promising for classification tasks comprising more classes. In our opinion oversizing the networks combined with early stopping is

the key for these encouraging results. Also Prechelt himself speculates that more than 32 hidden neurons (the maximum number he used) may produce superior results [12].

In future we will have to evaluate our results using further benchmarks and to analyse the effectiveness of each strategy in detail. Moreover we will add missing values strategies, feature selection, and some kind of knowledge extraction. When implemented all this in an intuitively applicable fashion, the basis will be done for a wide-spread use of neural network technique in numerous medical fields.

References

1. Rumelhart, D. E., Hinton, G. E., & Williams, R. J.: Learning Representations by Back-Propagating Errors. Nature 323 (1986) 533-536
2. Penny, W., Frost, D.: Neural Networks in Clinical Medicine. Med. Decis. Making 16 (1996) 386-398
3. Orr, G. B., Müller, K.-R. (eds.): Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, Vol. 1524. Springer-Verlag, Berlin Heidelberg New York (1998)
4. Kwok, T.Y., Yeung, D.Y.: Constructive algorithms for structure learning in feed forward neural networks for regression problems. IEEE Trans. on Neural Networks 8(3) (1997) 630-645
5. Ash, T.: Dynamic node creation in backpropagation networks. Connection Science 1(4) (1989) 365-375
6. Fahlman, S.E., Lebiere, C.: The cascade-correlation learning architecture. In: Touretzky, D.S. (ed.): Advances in Neural Information Processing Systems 2, Morgan Kaufmann, CA (1990) 524-532
7. Lehtokangas, M.: Modeling with constructive backpropagation. Neural Networks 12 (1999) 707-716
8. Yang, J. Honavar V.: Experiments with the cascade-correlation algorithm. Microcomputer Applications 17 (1998) 40-46
9. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S.: Efficient Classification for Multiclass Problems Using Modular Neural Networks. IEEE Trans. on Neural Networks. 6(1) (1995) 117-124
10. Linder, R., Wirtz, S., Pöppl, S.J.: Speeding up Backpropagation Learning by the APROP Algorithm. Second International ICSC Symposium on Neural Computation, Proceedings CD, Berlin (2000)
11. Linder, R., Pöppl, S.J.: Accelerating Backpropagation Learning: The APROP Algorithm. Neural Computation, submitted
12. Prechelt, L.: Proben 1 – a set of neural network benchmark problems and benchmarking rules. Technical Report 21/94, Fakultät für Informatik, Universität Karlsruhe. Available via ftp.ira.uka.de in directory /pub/papers/techreports/1994 as file 1994-21.ps.Z. (1994)
13. Qian, N., Sejnowski, T.J.: Predicting the Secondary Structure of Globular Proteins Using Neural Network Models. J. Mol. Biol. 202 (1988) 865-884
14. Amari, S., Murata, N., Müller, K., Finke, M., Yang, H.H.: Asymptotic statistical theory of overtraining and cross-validation. IEEE Trans. On Neural Networks 8(5) (1997) 985-996
15. Wang, C., Venkatesh, S.S., Judd, J.S.: Optimal stopping and effective machine complexity in learning. Advances in Neural Information Processing Systems 6 (1995) 303-310
16. Finnoff, W., Hergert, F., Zimmermann, H.G.: Improving model selection by nonconvergent methods. Neural Networks 6 (1993) 771-783

17. Prechelt, L.: Automatic early stopping using cross validation: quantifying the criteria. Neural Networks 11 (1998) 761-767
18. Riedmiller, M., & Braun, H.: A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm. In: Proc. I.E.E.E. International Conference on Neural Networks 1. San Francisco (1993) 586-591
19. Finnoff, W., Hergert, F., Zimmermann, H. G.: Neuronale Lernverfahren mit variabler Schrittweite. Tech. report, Siemens AG (1993)
20. Linder, R., Pöppl, S.J.: Backprop, RPROP, APROP: Searching for the best learning rule for an electronic nose. Neural Networks in Applications '99. In: Proc. of the Fourth International Workshop, Magdeburg, Germany, (1999) 69-74
21. Ng, S.C., Leung, S.H., LIK, A.: Fast Convergent Generalized Back-Propagation Algorithm with Constant Learning Rate. Neural Processing Letters 9 (1999) 13-23
22. Looney, C.G.: Stabilization and speedup of convergence in training feedforward neural networks. Neurocomputing 10 (1996) 7-31
23. Blumer, A., Ehrenfeucht, A., Haussler, D., Warmuth, M.: Occam´s razor. Information Processing Letters 24(6) (1987) 377-380
24. Weigend, A.: On overfitting and the effective number of hidden units. In: Proc. of the 1993 Connectionist Models Summer School (1994) 335-342
25. Sarle, W.S.: Stopped Training and Other Remedies for Overfitting. In: Proc. of the 27th Symposium on the Interface of Computer Science and Statistics (1995) 352-360
26. Lawrence, S., Giles C.L., Tsoi, A.C.: What size neural network gives optimal generalization? Convergence properties of backpropagation. Tech. Rep. No. UMIACS-TR-96-22 and CS-TR-3617. University of Maryland, College Park, MD 20742: Institute for Advanced Computer Studies (1996)
27. Caruana, R.: Generalization vs. Net Size. Neural Information Processing Systems, Tutorial, Denver, CO (1993)
28. Schraudolph, N.N.: A Fast, Compact Approximation of the Exponential Function. Neural Computation 11 (1999) 853-862
29. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.-R.: Efficient BackProp. In: Orr, G.B., Müller, K.-R. (eds.): Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, Vol. 1524. Springer-Verlag, Berlin Heidelberg New York (1998) 9-50
30. Becker, S., LeCun, Y.: Improving the Convergence of Back-Propagation Learning With Second Order Methods. In: Touretzky, D.S., Hinton, G.E., Sejnowski, T. (eds.): Proc. Of the 1988 Connectionist Models Summer School. Morgan Kaufmann, San Mateo, CA (1989) 29-37

Nonlinear Function Learning and Classification Using Optimal Radial Basis Function Networks

Adam Krzyżak*

Department of Computer Science, Concordia University
1455 de Maisonneuve Blvd. W.
Montreal Quebec, Canada H3G 1M8
krzyzak@cs.concordia.ca

Abstract. We derive optimal radial kernel in the radial basis function network applied in nonlinear function learning and classification.

1 Introduction

In this article we study the problem of nonlinear regression estimation and classification by the radial basis function (RBF) networks with k nodes and a fixed kernel $\phi : \mathcal{R}_+ \to \mathcal{R}$:

$$f_k(x) = \sum_{i=1}^{k} w_i \phi\left(||x - c_i||_{A_i}\right) + w_0 \tag{1}$$

where

$$\|x - c_i\|_{A_i}^2 = [x - c_i]^T A_i [x - c_i],$$

scalars $w_1 \ldots, w_k$, $c_1, \ldots, c_k \in \mathcal{R}^d$, positive semidefinite matrices $A_1, \ldots, A_k \in \mathcal{R}^d \times \mathcal{R}^d$ are parameters of the network and $\phi\left(||x - c_i||_{A_i}\right)$ is the radial basis function. A_i are covariance matrices and c_i are called centers. RBF networks have been introduced by Broomhead and Lowe [1] and Moody and Darken [15]. Their convergence in regression estimation problem and classification was studied by Krzyżak et al [12], rates of approximation by Park and Sandberg [16,17], Girosi and Anzellotti [7] and rates in nonlinear function estimation problem by McCaffrey and Gallant [14] and Krzyżak and Linder [13]. Typical forms of radial functions encountered in estimation applications are monotonically decreasing functions such as:

- $\phi(x) = e^{-x^2}$ (Gaussian kernel)
- $\phi(x) = e^{-x}$ (exponential kernel)
- $\phi(x) = (1 - x^2)_+$ (truncated parabolic kernel)
- $\phi(x) = \frac{1}{\sqrt{x^2 + c^2}}$ (inverse multiquadratic)

In approximation and interpolation [18] increasing kernels prevail. Some examples of these are:

* This research was supported by the Alexander von Humboldt Foundation and Natural Science and Engineering Research Council of Canada

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 217–225, 2001.

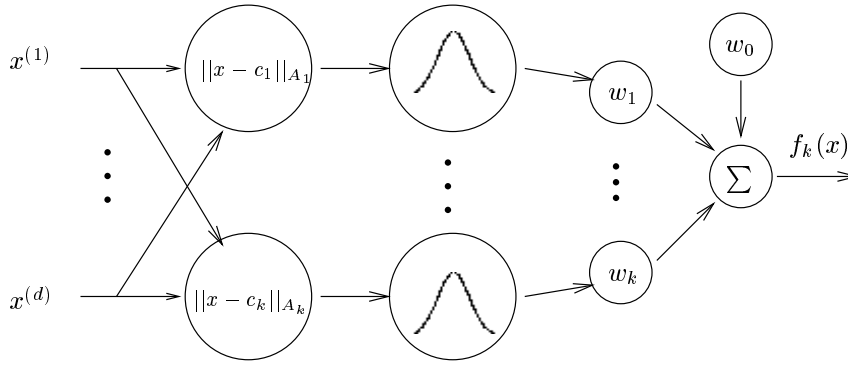


Fig. 1. Radial basis network with one hidden layer

- $\phi(x) = \sqrt{x^2 + c^2}$ (multiquadratic)
- $\phi(x) = x^{2n} \log x$ (thin plate spline)

The results obtained in this paper are motivated by the study of the optimal kernel in density estimation by Watson and Leadbetter [17] subsequently specialized to a class of Parzen kernels by Davis [2].

2 MISE Optimal RBF Networks for Known Input Density

In this section we will present the optimal RBF network in mean integrated square error sense (MISE) and the corresponding optimal rate of convergence in the regression estimation and classification problem. Let $(X, Y) \in \mathcal{R}^d \times \mathcal{R}$ be random vector and let probability density of X be known and denoted by $f(x)$. Let $E\{Y|X = x\} = R(x)$ be regression function of Y given X and $EY^2 < \infty$. In the sequel we will propose RBF network estimate of $G(x) = R(x)f(x)$. We will analytically minimize MISE of the estimate of G and obtain implicit formula for optimal kernel. We also obtain the exact expression for the MISE rate of convergence of the optimal RBF network estimate. Estimation of G is important in the following two situations:

1. Nonlinear estimation.
   Consider the model $Y = R(X) + Z$, where Z is zero mean noise and R is unknown mapping. We would like to approximate an unknown nonlinear input-output mappings R. Clearly R is the regression function $E(Y|X = x)$. In order to estimate R we generate a sequence of i.i.d. random variables $X_1, \ldots, X_n$ from X, whose density is known (e.g. uniform on the interval on which we want to reconstruct R) and observe $Y's$. We construct estimate G_n of G. The estimate enables us to recover $G(x) = R(x)f(x)$. Hence the estimate of R is trivially given by $G_n(x)/f(x)$.

2. Classification.
In the classification (pattern recognition) problem, we try to determine a label Y corresponding to a random feature vector $X \in \mathcal{R}^d$, where Y is a random variable taking its values from $\{-1, 1\}$. The decision function is $g : \mathcal{R}^d \to \{-1, 1\}$, and its goodness is measured by the *error probability* $L(g) = \mathbf{P}\{g(X) \neq Y\}$. It is well known that the decision function that minimizes the error probability is given by

$$g^*(x) = \begin{cases} -1 \text{ if } R(x) \leq 0 \\ 1 \quad \text{otherwise,} \end{cases}$$

where $R(x) = \mathbf{E}(Y|X = x)$, g^* is called the *Bayes decision*, and its error probability $L^* = \mathbf{P}\{g^*(X) \neq Y\}$ is the *Bayes risk*.
When the joint distribution of (X, Y) is unknown (as is typical in practical situations), a good decision has to be learned from a training sequence

$$D_n = ((X_1, Y_1), \ldots, (X_n, Y_n)),$$

which consists of n independent copies of the $\mathcal{R}^d \times \{-1, 1\}$-valued pair (X, Y). Then formally, a decision rule g_n is a function $g_n : \mathcal{R}^d \times (\mathcal{R}^d \times \{-1, 1\})^n \to \{-1, 1\}$, whose error probability is given by

$$L(g_n) = \mathbf{P}\{g_n(X, D_n) \neq Y|D_n\}.$$

Note that $L(g_n)$ is a random variable, as it depends on the (random) training sequence D_n. For notational simplicity, we will write $g_n(x)$ instead of $g_n(x, D_n)$.
A sequence of classifiers $\{g_n\}$ is called *strongly consistent*, if

$$\mathbf{P}\{g_n(X) \neq Y|D_n\} - L^* \to 0 \text{ almost surely (a.s.) as } n \to \infty,$$

and $\{g_n\}$ is *strongly universally consistent* if it is consistent for *any* distribution of (X, Y).
Pattern recognition is closely related to regression function estimation. This is seen by observing that the function R defining the optimal decision g^* is just the regression function $\mathbf{E}(Y|X = x)$. Thus, having a good estimate $R_n(x)$ of the regression function R, we expect a good performance of the decision rule

$$g_n(x) = \begin{cases} -1 \text{ if } R_n(x) \leq 0 \\ 1 \quad \text{otherwise.} \end{cases} \tag{2}$$

Indeed, we have the well-known inequality

$$\mathbf{P}\{g_n(X) \neq Y|X = x, D_n\} - \mathbf{P}\{g^*(X) \neq Y|X = x\} \leq |R_n(x) - R(x)|$$

(see e.g. Devroye et al [6]), and in particular,

$$\mathbf{P}\{g_n(X) \neq Y|D_n\} - \mathbf{P}\{g^*(X) \neq Y\} \leq \left(\mathbf{E}\left((R_n(X) - R(X))^2 \middle| D_n\right)\right)^{1/2}.$$

Therefore, any strongly consistent estimate R_n of the regression function R leads to a strongly consistent classification rule g_n via (2). For example, if R_n is an RBF-estimate of R based on minimizing the empirical L_2, then according to the consistency theorem discussed in the previous section, g_n is a strongly universally consistent classification rule. That is, for any distribution of (X, Y), it is guaranteed that the error probability of the RBF-classifier gets arbitrarily close to that of the best possible classifier if the training sequence D_n is long enough.

Bayes rule can also obtained by assigning a given feature vector x to a class with the highest *a posteriori* probability, i.e. by assigning x to class i, if $P_i(x) = \max_j P_j(x)$, where $P_i(x) = EI_{(\theta=i, X=x)} = ER_i(X)$, θ is a class label, $R_i(x) = EI_{(\theta=i|X=x)}$ and I_A is indicator of set A [10,11]. Therefore it is essential to estimate G to obtain a good classification rule.

Suppose that $(X_1, Y_1), \cdots, (X_n, Y_n)$ is a sequence of i.i.d. observations of (X, Y). Consider generalization of (1) by allowing each radial function $\phi\left(||x - c_i||_{A_i}\right)$ to depend on k, where $||\cdot||$ is an arbitrary norm (not necessary Euclidean). Thus

$$f_k(x) = \sum_{i=1}^{k} w_i \phi_k \left(||x - c_i||_{A_i}\right) + w_0. \tag{3}$$

There are several approaches to learn parameters of the network. In empirical risk minimization approach the parameters of the network are selected so that empirical risk is minimized, i.e.

$$J_n(f_\theta) = \min_{\hat{\theta} \in \Theta_n} J_n(f_{\hat{\theta}}).$$

where

$$J_n(f_\theta) = \frac{1}{n} \sum_{j=1}^{n} |f_\theta(X_j) - Y_j|^2$$

is the empirical risk and

$$\Theta_n = \left\{ \theta = (w_0, \ldots, w_{k_n}, c_1, \ldots, c_{k_n}, A_1, \ldots, A_{k_n}) : \sum_{i=0}^{k_n} |w_i|^2 \leq b_n \right\},$$

is the vector of parameters. In order to avoid too close fit of the network to the data (overfitting problem) we carefully control the complexity of the network expressed by the number of hidden units k as the size of the training sequence increases. This is the method of sieves of Grenander [8]. The complexity of the network can also be described by the Vapnik-Chervonenkis dimension [6]. This approach has been applied to learning of RBF networks in [12]. The learning is consistent for bounded output weights and the size of the network increasing according to the condition $k_n^3 b_n^2 \log(k_n^3 b_n^2)/n \to 0$ as $n \to \infty$. It means that the network performs data compression.

Empirical risk minimization is asymptotically optimal strategy but it has high computational complexity. A simpler parameter training approach consists

of assigning data values to output weights and centers (plug-in approach). This approach does not offer compression but is easy to implement. Consistency of plug-in RBF networks was investigated in [20].

In this paper we focus our attention on plug-in approach. Let parameters of (3) be trained as follows:

$$k_n = n, w_0 = 0, w_i = Y_i,\ c_i = X_i,\ i = 1, \cdots, n.$$

Consider kernel $K : \mathcal{R}^d \to \mathcal{R}$ which is radially symmetric $K(x) = K(||x||)$. Network (3) can be rewritten

$$f_k(x) = G_n(x) = \frac{1}{n}\sum_{i=1}^{n} Y_i K_n(x - X_i).$$

We define RBF estimate of G by

$$G_n(x) = \frac{1}{n}\sum_{i=1}^{n} Y_i K_n(x - X_i)$$

where $K_n(x)$ is some square integrable kernel. We consider MISE of $G_n(x)$

$$Q = E\int \left(G(x) - G_n(x)\right)^2 dx \tag{4}$$

where $\int$ is taken over $\mathcal{R}^d$. We are interested in deriving an optimal kernel K_n^* minimizing Q.

For the sake of simplicity in the remainder of the paper we only consider scalar case $d = 1$. In what follows we will use the elements of Fourier transform theory [9]. Denote by Φ_g Fourier transform of g, i.e.

$$\Phi_g(t) = \int g(x)e^{itx}dx$$

and thus inverse Fourier transform is given by

$$g(x) = \frac{1}{2\pi}\int \Phi_g(t)e^{-itx}dt.$$

The optimal form of K_n is given in Theorem 1.

Theorem 1. *The optimal kernel K_n^* minimizing (4) is defined by the equation*

$$\Phi(K_n^*) = \frac{n|\Phi_G|^2}{EY^2 + (n-1)|\Phi_G|^2}. \tag{5}$$

The optimal rate of MISE corresponding to the optimal kernel (5) is given by

$$Q_n^* = \frac{1}{2\pi}\int \frac{\left(EY^2 - |\Phi_G(t)|^2\right)|\Phi_G(t)|^2}{EY^2 + (n-1)|\Phi_G(t)|^2}dt.$$

Observe that

$$Q_n^* = \frac{EY^2}{2\pi}\int \frac{|\Phi_G(t)|^2}{EY^2+(n-1)|\Phi_G(t)|^2}dt$$
$$-\frac{1}{2\pi}\int \frac{|\Phi_G(t)|^4}{EY^2+(n-1)|\Phi_G(t)|^2}dt$$
$$\leq \frac{EY^2K_n^*(0)}{n} - \frac{1}{2\pi(n-1)}\int |\Phi_G(t)|^2dt.$$

Kernel (5) is related to the superkernel of [3,4]. Notice that for band-limited R and f with the rate of MISE convergence with kernel (5) is

$$nQ_n^* \to \frac{1}{2\pi}\int_{-T}^{T}\left(EY^2 - |\Phi_G(t)|^2\right)dt \tag{6}$$

as $n \to \infty$, where T is the maximum of bands of R and f. Therefore $Q_n^* = O(1/n)$. For other classes of R and f we will get different optimal kernels and rates. To get an idea how the optimal kernel may look like consider $G(x) = e^{-x}$. It can be shown that

$$K_n^*(x) = \frac{n}{EY^2}\sqrt{\frac{\pi EY^2}{2EY^2+2(n-1)}}\exp(-\sqrt{1+\frac{n-1}{EY^2}}|x|)$$

so by the formula (6) we have

$$nQ_n^* = \sqrt{\frac{\pi EY^2 n}{2EY^2+2(n-1)}} - \frac{\sqrt{n}}{2\pi(n-1)}\int |\Phi_G(t)|^2dt \to \sqrt{\frac{\pi EY^2}{2}}$$

$|t| \to \infty$.We next consider polynomial classes and exponential classes that is classes of R and f with tails of Φ_R and Φ_f decreasing either polynomially or exponentially. The rate of decrease affects the shape of optimal kernel and the optimal rate of convergence of MISE.

We say that Φ_G has **algebraic rate** of decrease if

$$|t|^p|\Phi_G| \to \sqrt{K}$$

as $|t| \to \infty$.

One can show that

$$n^{1-1/2p}Q_n^* \to \frac{1}{2\pi}\left(\frac{K}{EY^2}\right)^{1/2p}\int \frac{dx}{1+|x|^{2p}}$$

for $p > 1/2$.

Φ_G has **exponential rate** of decrease if

$$|\Phi_G| \leq Ae^{\rho|t|}$$

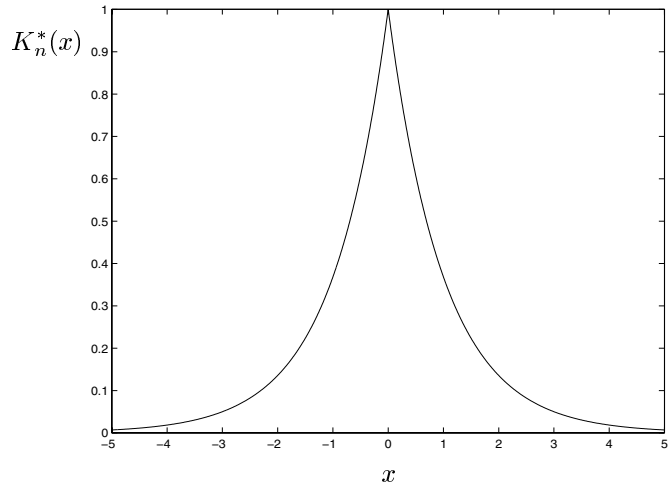


Fig. 2. Optimal RBF kernel

for some constant A, all t and $\rho > 0$ and

$$\lim_{s\to\infty} \int_0^1 [1 + e^{2\rho s} |\Phi_G(st)|^2] dt = 0.$$

It can be shown

$$\frac{n}{\log n} Q_n^* = \frac{1}{2\pi} \frac{n}{\log n} \int \frac{|\Phi_G(t)|^2}{EY^2 + (n-1)|\Phi_G(t)|^2} dt \to \frac{EY^2}{2\pi\rho}$$

as $n \to \infty$.

3 MISE Optimal RBF Networks for Unknown Input Density

In this section we will derive the optimal RBF regression estimate and the optimal rate of convergence in case when density f is estimated from $(X_1, Y_1), \cdots, (X_n, Y_n)$. Consider RBF regression estimate

$$R_n(x) = \frac{G_n(x)}{f_n(x)} = \frac{\frac{1}{n}\sum_{i=1}^n Y_i K_n(x - X_i)}{\frac{1}{n}\sum_{i=1}^n K_n(x - X_i)}.$$

Instead of working with MISE directly we will use the following relationship

$$EX \le \epsilon + (M - \epsilon) P\{X > \epsilon\}$$

which provides for bounded random variables the upper bound for MISE in terms of the probability of deviation provided that $X \le M$. We will optimize

the upper bound with respect to K_n. Let

$$Q = E \int |R_n(x) - R(x)|^2 f(x) dx. \tag{7}$$

The next theorem gives the form of the radial function minimizing the bound on Q and the corresponding upper bound on the rate of convergence.

Theorem 2. *The optimal kernel K_n^* minimizing upper bound on (7) is defined by the equation*

$$\Phi(K_n^*) = \frac{n[|\Phi_f|^2 + |\Phi_G|^2]}{(1 + EY^2) + (n-1)[|\Phi_f|^2 + |\Phi_G|^2]}. \tag{8}$$

The optimal rate of MISE corresponding to the optimal kernel (8) is given by

$$\bar{Q}_n^* = \frac{1}{2\pi} \int \frac{\left(EY^2 - [|\Phi_f(t)|^2 + |\Phi_G(t)|^2]\right) |\Phi_G(t)|^2}{(1 + EY^2) + (n-1)[|\Phi_f(t)|^2 + |\Phi_G(t)|^2]} dt.$$

The optimal radial function depends on density of X and on regression function or a posteriori class probability G. It is clear that MISE rate of convergence of RBF net in nonlinear learning problem with band-limited f and G is $1/n$. For classes of functions with algebraic or exponential rate of decrease we get similar rates as in the previous section. Observe that we achieve a parametric rate in intrinsically non-parametric problem.

The comparison of the performance of RBF networks in classification problem with standard and optimal radial functions will be left for future work.

References

1. D. S. Broomhead, D. Lowe, Multivariable functional interpolation and adaptive networks, *Complex Systems* 2 (1988) 321-323.
2. K. B. Davis, Mean integrated error properties of density estimates, *Annals of Statistics* 5, (1977) 530-535.
3. L. Devroye, A note on the usefulness of superkernels in density estimation, *Annals of Statistics* 20 (1993) 2037-2056.
4. L. Devroye, *A Course in Density Estimation*, Birkhauser, Boston, 1987.
5. L. Devroye, A. Krzyżak, An equivalence theorem for L_1 convergence of the kernel regression estimate, *J. of Statistical Planning and Inference* 23 (1989) 71-82.
6. L. Devroye, L. Györfi and G. Lugosi, *Probabilistic Theory of Pattern Recognition*, Springer, 1996.
7. F. Girosi, G. Anzellotti, Rates of converegence for radial basis functions and neural networks, in: R. J. Mammone (Ed.), *Artificial Neural Networks for Speech and Vision*, Chapman and Hall, London, 1993, 97-113.
8. U. Grenander. *Abstract Inference*. Wiley, New York, 1981.
9. T. Kawata, Fourier *Analysis in Probability Theory*, Academic Press, New York, 1972.
10. A. Krzyżak, The rates of convergence of kernel regression estimates and classification rules, *IEEE Trans. on Information Theory* 32 (1986) 668-679.

11. A. Krzyżak, On exponetial bounds on the Bayes risk of the kernal classification rule, *IEEE Transactions on Information Theory* 37 (1991) 490-499.
12. A. Krzyżak, T. Linder, G. Lugosi, Nonparametric estimation and classification using radial basis function nets and empirical risk minimization, *IEEE Transactions on Neural Networks* 7 (1996) 475–487.
13. A. Krzyżak, T. Linder, Radial Basis Function Networks and Complexity Regularization in Function Learning, *IEEE Transactions on Neural Networks* 9 (1998) 247–256.
14. D. F. McCaffrey, A. R. Gallant, Convergence rates for single hidden layer feedforward networks, *Neural Networks* 7 (1994) 147–158.
15. J. Moody, J. Darken, Fast learning in networks of locally-tuned processing units, *Neural Computation* 1 (1989) 281-294.
16. J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, *Neural Computation* 3 (1991) 246-257.
17. J. Park, I. W. Sandberg, Universal approximation using radial-basis-function networks, *Neural Computation* 5 (1993) 305-316.
18. M. J. D. Powell, Radial basis functions for multivariable approximation: a review, in J. C. Mason and M. G. Cox (Eds.), *Algorithms for Approximation*, Oxford University Press, 1987, 143–167.
19. G. S. Watson, M. R. Leadbetter, On the estimation of the probability density, I, *Annals of Mathematical Statistics* 34 (1963) 480-491.
20. L. Xu, A. Krzyżak, A. L. Yuille, On Radial Basis Function Nets and Kernel Regression: Statistical Consistency, Convergence Rates and Receptive Field Size, *Neural Networks* 7 (1994) 609-628.

Local Learning Framework for Recognition of Lowercase Handwritten Characters

Jian-xiong Dong[1], Adam Krzyżak[2], and C.Y. Suen[1]

[1] Centre Of Pattern Recognition and Machine Intelligence
Concordia University
Montreal Quebec, Canada H3G 1M8
{jdong, suen}@cenparmi.concordia.ca
[2] Department of Computer Science, Concordia University
1455 de Maisonneuve Blvd. W.
Montreal Quebec, Canada H3G 1M8
krzyzak@cs.concordia.ca

Abstract. This paper proposes a general local learning framework to effectively alleviate the complexities of classifier design by means of "divide and conquer" principle and ensemble method. The learning framework consists of quantization layer and ensemble layer. After GLVQ and MLP are applied to the framework, the proposed method is tested on public handwritten lowercase data sets, which obtains a promising performance consistently. Further, in contrast to LeNet5, an effective neural network structure, our method is especially suitable for a large-scale real-world classification problem although it is easily scaled to a small training set with preserving a good performance.

1 Introduction

Over the last decade, neural networks have been gradually applied to solve very complex classification problems in the real world. There is a growing realization that these problems can be facilitated by the development of multi-net systems [1]. Multi-net systems can provide feasible solutions to some difficult tasks that can not be solved by a single net. A single neural net often exhibits the overfitting problem which results in a weak generalization performance when trained on a limited set of training data. Some theoretical and experimental results [2], [3] have shown that an ensemble of neural networks can effectively reduce the variance that is directly related to the classification error.

A number of studies have addressed the problems of the construction of a multi-net system to achieve a better performance. The ensemble ("committee") and modular combination are two basic methods to construct multi-net systems. The two popular ensemble methods are Bagging [4] and AdaBoost [5]. Bagging employs the bootstrap sampling method to generate training subsets while the creation of each subset in AdaBoost depends on previous classification results. Compared with Bagging, AdaBoost obviously attempts to capture the classification information of "hard" patterns. However, its disadvantage is that it easily

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 226–238, 2001.

fits the noise in the training data. For modular combination, the task or problem is decomposed into a number of subtasks, and a complete task solution requires the contribution of all the modules [1]. Jocobs [6] proposed a mixture-of-experts model that consists of expert networks and a gating network. The training goal is to have the gating network learn an appropriate decomposition of the input space into different regions and switch the most suitable expert network to generate the outputs of input vectors falling within each region. In the model, the assumption of gaussian distribution in a local region is adopted, which does not make sense for a complex data distribution. Further, the model only selects the most suitable expert network to make a decision, rather than combining decisions of different expert networks. Most experiments show that an ensemble method in a local region is more effective than the individual best neural network.

In this paper, we present a method to construct a hierarchical local learning framework for pattern classification to systematically address the above problems. The framework consists of two layers. In the first layer, the technique of Learning Vector Quantization (LVQ) is used to partition a pattern space into clusters or subspaces. In the second layer, different ensembles of local learning machines that are trained in neighboring local regions are employed to make decision for classification. LVQ, which minimizes the average expected misclassification error, builds piecewise linear hyperplanes between neighboring codebook vectors to approximate Bayes decision boundary [7]. Due to the complex decision boundary for real world classification problems, there exists approximation error for piecewise linear hyperplanes. Therefore, in order to better approximate Bayes decision boundary, more powerful neural network ensembles are used to fine-tune it and reduce the prediction error.

The remainder of the paper is organized as follows. First, the learning framework is presented, followed by a consideration of how to choose the right models. In section 4, experimental results on handwritten NIST and CENPARMI lowercase databases are provided to illustrate the advantage of the proposed method. Finally, conclusions are drawn in section 5.

2 Formulation of Learning Framework

Before we select the specified models, it seems appropriate to formally define each part of the proposed learning framework. Fig. 1 shows a basic structure of the system that consists of a vector quantization layer and an ensemble layer.

2.1 Vector Quantization

Vector quantization can be considered as a method of signal compression at low cost where most information is reproduced in the number of codebook vectors. The traditional mean squared error (MSE) is often assumed as a design criterion. For labeled patterns, the limits of those approaches based on this criterion are that an accurate representation of the observation vector in terms of MSE may not lead to an accurate reproduction of Bayes rule [8]. Under Bayes decision

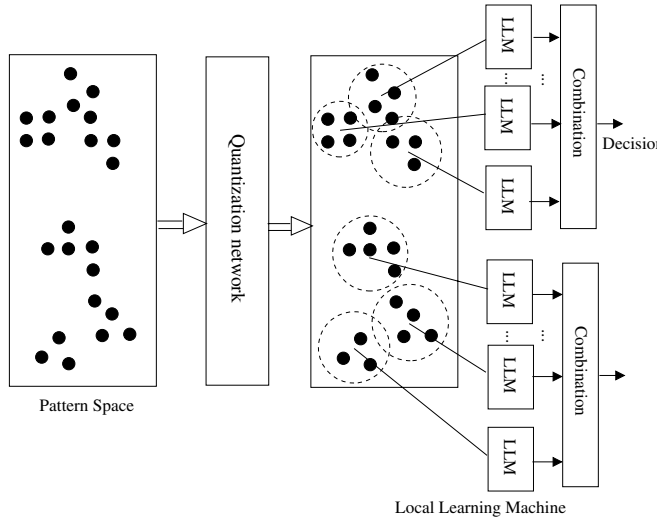


Fig. 1. A general local learning framework. Here local learning machines denote classifiers that are designed in local regions

philosophy, Kohonen intuitively introduced LVQ1, where information about the class to which a pattern belongs is exploited [9]. Kohonen's LVQ1, which is not derived from an explicit cost function, has been shown that it does not minimize the Bayes risk [10]. In order to address the problems, Juang & Katagiri [11] proposed an effective discriminative learning criterion called Minimum Classification Error (MCE) that minimizes the expectation loss in Bayes decision theory by a gradient descent procedure. Several generalized LVQs based on MCE [12], [13], [14] were proposed. Here we unify them in a consistent framework.

Let m_{kr} be the r-th reference vector in class w_k. $\Lambda_k = \{m_{kr} | r = 1, \cdots, n_k\}$, $k = 1, \cdots, K$ where K is the number of classes, and $\Lambda = \bigcup_{k=1}^{K} \Lambda_k$. Suppose that input vector $x (\in w_k)$ is presented to the system. Let $g_k(x; \Lambda)$ denote the discriminant function of class w_k as follows:

$$g_k(x; \Lambda) = \varphi(x; \Lambda_k) \ . \tag{1}$$

where $\varphi(x; \Lambda_k)$ is a smooth function. The misclassification measure, denoted by $\mu_k(x; \Lambda)$, has the following form:

$$\mu_k(x; \Lambda) = \frac{-g_k(x; \Lambda) + [\frac{1}{K-1} \sum_{j, j \neq k} g_j(x; \Lambda)^\eta]^{\frac{1}{\eta}}}{g_k(x; \Lambda) + [\frac{1}{K-1} \sum_{j, j \neq k} g_j(x; \Lambda)^\eta]^{\frac{1}{\eta}}} \ . \tag{2}$$

where $g_j, j = 1, \cdots, K$ are assumed to be positive[1]. Note that for a sufficiently large value of η, $\mu_k(x; \Lambda) \leq 0$ implies misclassification and $\mu_k(x; \Lambda) \geq 0$ corresponds to the correct decision.

In Bayes decision theory, we often minimize a risk function to evaluate the decision results. In order to make loss function differentiable, we take into account a "soft" nonlinear sigmoid function instead of a "hard" zero-one threshold.

$$\begin{aligned} l_k(x; \Lambda) &= l_k(\mu_k(x; \Lambda)) \\ &= \frac{1}{1+\exp(-\xi(t)\mu_k(x;\Lambda))} \quad (\xi > 0) \ . \end{aligned} \tag{3}$$

Thus an empirical loss can be expressed by

$$L(\Lambda) = \frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} l_k(x_i; \Lambda) 1_{(x_i \in w_k)} \ . \tag{4}$$

where N is the number of training samples and $1_{(.)}$ is an indicator function. Then the cost function can be minimized by a gradient descent procedure denoted by

$$\Lambda_{t+1} \leftarrow \Lambda_t - \epsilon(t) \nabla L(\Lambda_t) \ . \tag{5}$$

where Λ_t denotes the parameters set at the tth iteration and ϵ is the learning rate.

2.2 Construction of Ensembles

After vector quantization, each reference vector can be regarded as a cluster center. Although we can collect training subsets for each cluster by the nearest neighboring rule and train local learning machines on these subsets, there exist the following limits:

- Training samples on some subsets are insufficient; as a result, classifiers designed on these subsets will have a weak generalization ability.
- This method ignores information of "boundary patterns" between neighboring clusters while most misclassification errors occur on these boundaries.

In order to overcome the above problems, we inject neighboring samples into training subsets. The procedure is illustrated in Fig. 2.

From the above procedure, we can observe that the obtained sets S_j are partially overlapping. Finally, we build the ensemble of networks naturally in the Bayesian framework. We employ neural networks to model the *posteriori* probability by the mixture of the neighboring expert nets. That is,

$$P(w_k|x) = \sum_{i=1}^{L} P(e_i) P(w_k|x, e_i) \ . \tag{6}$$

Here we assume that each expert net e_i is independent. $P(e_i)$ denotes a *priori* probability of expert net e_i and $P(w_k|x, e_i)$ means a *posteriori* probability for expert net e_i.

[1] If g_j are negative and bounded, add a sufficiently large positive constant M to g_j such that $M + g_j \ j = 1, \cdots, K$ are positive.

Collect training subsets
Input: A series of training samples $x_1, x_2, \cdots, x_N$, where N is the number of samples, and reference vectors m_i, $i = 1, \cdots, \sum_{k=1}^{K} n_k$ where K and n_k denote the number of classes and the number of reference vectors for class w_k respectively, and sets S_j, $j = 1, \cdots, \sum_{k=1}^{K} n_k$.
Output: training subsets S_j, $j = 1, \cdots, \sum_{k=1}^{K} n_k$.
Initialize: Set sets S_j to be empty.
for $p = 1$ to N
1. For sample x_p, find L nearest neighboring reference vectors. That is,

$$i_1 = \arg\min_i \parallel x_p - m_i \parallel$$
$$i_k = \arg \min_{i \notin \{i_1, i_2, \cdots, i_{k-1}\}} \parallel x_p - m_i \parallel$$

where $\parallel . \parallel$ denotes the Euclidean norm and $k = 1, \cdots, L$.
2. Inject sample x_p into L training subsets

$$S_j = \{x_p\} \bigcup S_j \quad j \in \{i_1, i_2, \cdots, i_L\}$$

end **for**

Fig. 2. Procedure for collecting training subsets

3 Model Selection

The framework introduced in section 2 can be applied to different learning models. Thus model selection plays an important role in the overall performance of the designed system. For vector quantization, although some clustering models such as self-organizing maps [7] are available, these models have a common characteristic. That is, they use the MSE criterion, which is not directly related to the minimization of Bayes risk.

Based on the above arguments and discussion in section 1, we select generalized learning vector quantization proposed by Sato & Yamada [13], [14]. They showed that convergence property of reference vectors depends on the definition of the misclassification measure and their definition guarantees the convergence [13], [14], [15]. In their definition, the discriminant function (see eq. (1)) can be defined by the squared Euclidean distance as $g_k(x; \Lambda) = -d_k = -\min_r \parallel x - m_{kr} \parallel^2 = - \parallel x - m_{ki} \parallel^2$. In addition, as $\eta \to \infty$, equation (2) can be rewritten as

$$\begin{aligned} \mu_k(x; \Lambda) &= \frac{-g_k(x;\Lambda)+g_l(x;\Lambda)}{g_k(x;\Lambda)+g_l(x;\Lambda)} \\ &= \frac{d_k - d_l}{d_k + d_l} \ . \end{aligned} \tag{7}$$

where $g_l(x; \Lambda) = \max_{i \neq k} g_i(x; \Lambda) = \max_{i \neq k}[-\min_r \parallel x - m_{ir} \parallel^2] = -d_l = - \parallel x - m_{lj} \parallel^2$. Then according to equation (5), learning rules are

$$\begin{aligned} m_{ki} &\leftarrow m_{ki} + 4\epsilon(t)\xi(t)l(\mu_k)(1 - l(\mu_k))\frac{d_l}{(d_k+d_l)^2}(x - m_{ki}) \\ m_{lj} &\leftarrow m_{lj} - 4\epsilon(t)\xi(t)l(\mu_k)(1 - l(\mu_k))\frac{d_k}{(d_k+d_l)^2}(x - m_{lj}) \ . \end{aligned} \tag{8}$$

The above GLVQ assumes that initial positions and number of reference vectors of each class are known while they are unknown in practice. It is well known that initial positions of reference vectors have a great impact on the final performance of some clustering algorithms such as self-organizing maps, LVQ and neural gas. For LVQ, the traditional method is to employ k-means algorithm in the training data of each class to obtain the initial positions of the reference vectors. However, the classical k-means algorithm often converges to a "bad" local minimum. Further, a high computation cost also becomes a bottleneck of k-means algorithm for a large-scale clustering. Here we use an algorithm proposed by Bradley & Fayyad [16], which uses k-means algorithm and a "smoothing" procedure to refine the initial points. This algorithm is especially suitable for a large-scale clustering.

In the ensemble layer, we select multi-layer proceptrons (MLP) as local learning machines rather than support vector machine (SVM) although SVM's generalization power can be controlled more easily owing to two reasons. One is that MLP has a powerful nonlinear decision capability and is easy to implement. The other is that when training data are sufficiently large and the network is trained by minimizing a sum-of-square error function, the network output is the conditional average of the target data. When the target vector is one-of-place code, outputs can be regarded as *a posteriori* probability.

Finally, we use a simple averaging method to combine component expert nets because *a priori* probability of each expert net is unknown.

4 Experimental Results

In this section, we test our learning framework on several public handwritten character data sets, provide detailed experimental results and outline some related design parameters and discuss some issues of practical implementation. In addition, we provide an extensive performance comparison with other popular classifiers.

In our experiment, linear normalization and feature extraction based on the stroke edge are applied. All character images are size-normalized to fit the 32×32 box while preserving their aspect ratios. Also, a directional feature based on the gradient of gray scale image [17] is extracted by using the Robert edge operator. After a 400-dimensional feature vector has been extracted, principal component analysis can be employed to compress the high dimensional feature vector to a vector with 160 dimensions.

Before we present the experimental results, some related design parameters are provided. For GLVQ, we first use Bradly's algorithm [16] to determine the initial positions of twelve reference vectors within the training data of each class. In equation (3), $\xi(t)$ is set to be $(t/T+\xi_0)$ rather than t^2 recommended by Sato [14], where t refers to the number of presented samples and T denotes the number of training samples. ξ_0 is set to 0.05. The reason is that since the classification rate on the training rate is already high ($> 90\%$) after the initialization of GLVQ,

[2] Here t means the number of epochs that refers to inputting all training samples once.

a too small value of ξ_t results in a large penalty, which causes reference vectors to be adjusted dramatically. The learning step size decreases linearly with the number of steps t, i.e, $\alpha(t) = \alpha_0 \times (1.0 - t/t_{\max})$ with $\alpha_0 = 0.01$, where $t_{\max}$ is equal to epoch times the number of training sample. The epoch is set to 200. For the ensemble layer, multi-layered perceptrons (MLP) with a single hidden layer with 40 units are used as local learning machines. The sigmoid activation function is used, i.e, $1.0/(1.0+\exp(-x))$. All MLP's are trained using the gradient method with a momentum factor. The momentum factor and initial learning step size are set to 0.9 and 0.25, respectively. The number of component expert nets (L) is set to 15.

4.1 NIST Lowercase Database

NIST database of handwritten lowercase characters consists of 26,000 training samples and 12,000 test samples. In the training set, each category has 1000 samples. Since there are some garbages[3] and very confusing patterns such as "q" and "g", "i" and "l", where some patterns can be barely identified by human[4] and are shown in Fig. 3, we clean the database and just discard test samples of three categories including "q","i" and "g". Consequently, we obtain a training set with 23,937 samples and a test set with 10,688 samples.

Automatic recognition of handwritten lowercase characters without context information is a challenging task. In the past ten years, handwritten character recognition has made a great progress, especially in the online character and offline digit recognition. A quick scan of the table of contents of *IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE Transactions on Neural Networks, IEEE Transactions on Systems, Man, and Cybernetics, Pattern Recognition, International Journal of Pattern Recognition and Artificial Intelligence, Pattern Recognition Letters, The International Workshop on Frontiers in Handwriting Recognition and The International Conference on Document Analysis and Recognition* since 1990s reveals that little work has been done in handwritten lowercase recognition. There is no benchmark to compare different algorithms on the same database. Srihari [18] extracted some structural features such as 4-directional concaves, strokes (horizontal,vertical and diagonal), end-points, cross-points using morphological operators and three moment features and implemented a neural network classifier trained on NIST lowercase training subset with 10,134 samples using the above feature. The recognition rate on a modified NIST lowercase test set with 877 samples was 85%. Toshihiro [19] extracted and combined three different features that consist of stroke/background and contour-direction features. The proposed classifier is a three-layer MLP network trained on NIST training subset with 10,968 samples. The recognition rate for lowercase characters on the modified NIST test subset with 8,284 samples was 89.64%. Obviously, the above researchers discarded some

[3] The database contains some uppercase patterns and noisy patterns that do not belong to one of 26 categories.

[4] In the test set, about 6% patterns can not be identified by human

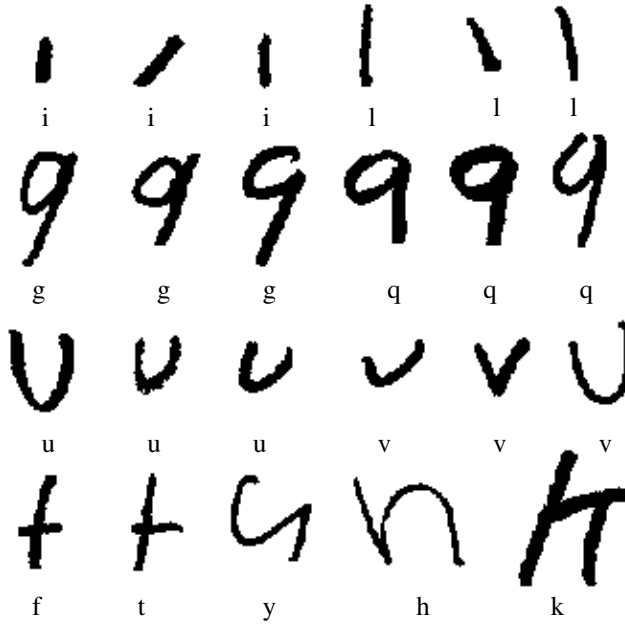


Fig. 3. Confusing patterns in NIST lowercase database

samples of the original test set and got a smaller subset. In summary, the above experimental results indicate that techniques of recognizing handwritten characters are far away from maturity. They differ from handwritten digit recognition in that there exists a great overlap between class pattern space. Similar patterns are distributed as clusters. This also motivates the usage of local ensembles to capture the discrimitive information of "boundary" patterns.

In the experiment, we pick up confusing patterns from the categories "g" and "q" and put them into a new category. Twenty-two classes are assigned to eight prototype vectors and five classes with a small number of data to four prototype vectors. The MLP in the ensemble layer contains 40 hidden units. The experimental results are illustrated in Fig. 4.

In practical handwriting recognition that integrates segmentation and classification or make use of postprocessing, the classifier does not necessarily output a unique class. The cumulative accuracy of top ranks is also of importance. Table 1 shows the cumulative recognition rate of the proposed method.

Table 1. Cumulative recognition rate of the proposed method (%)

Candidate	top rank	2 ranks	3 ranks	4 ranks	5 ranks
recognition rate	92.34%	96.9%	98.09%	98.46%	98.85%

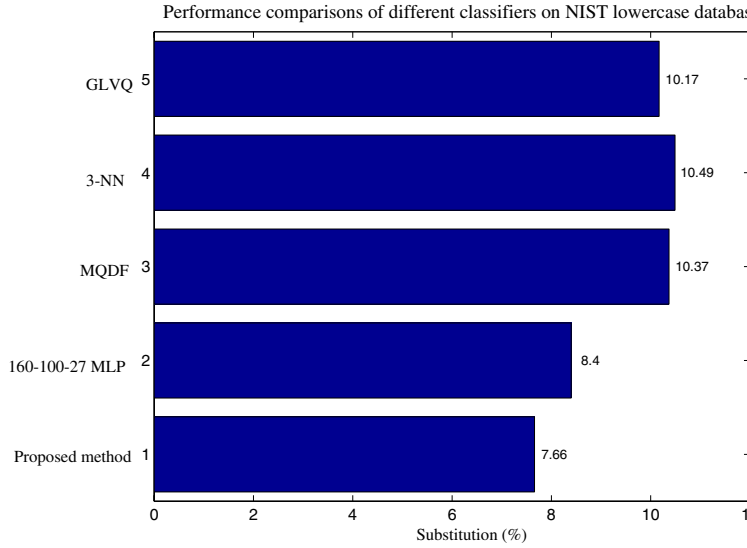


Fig. 4. Error rates of different methods on the test set of NIST lowercase database, each bar represents a classifier

4.2 CENPARMI Lowercase Database

Due to some problems with NIST lowercase database, we collected samples from university students and built a lowercase database. All samples are clean and preserve a complete stroke structure. But patterns within the same category have a variety of shapes. The lowercase samples are stored in bi-level format, whose scanning resolution is 300DPI. The database contains samples from 195 writers. We divide samples into training set and test set according to writer identities. The samples of randomly selected 120 writers are used as training set and the rest as test set. As a result, the training set consists of 14,810 samples and test set contains 8,580 samples. Fig. 5 shows some samples in the database.

In this experiment, we not only evaluate the performance of the proposed method but also investigate several factors that have an effect on the overall performance. First, we outline the designed parameter setting. The number of reference vectors of each class is set to eight and MLP in the ensemble layer has forty hidden units. Other parameters are the same as those in the first experiment. In order to better evaluate the performance, several other classifiers including boosted MLP are designed as a comparison with the proposed method. The experimental results are depicted in Fig. 6.

It can be observed that the proposed method outperforms the boosted MLP. AdaBoost is not as powerful as we expected. It almost does not boost the MLP classifier although training error is reduced to zero by combining fifteen component MLPs.

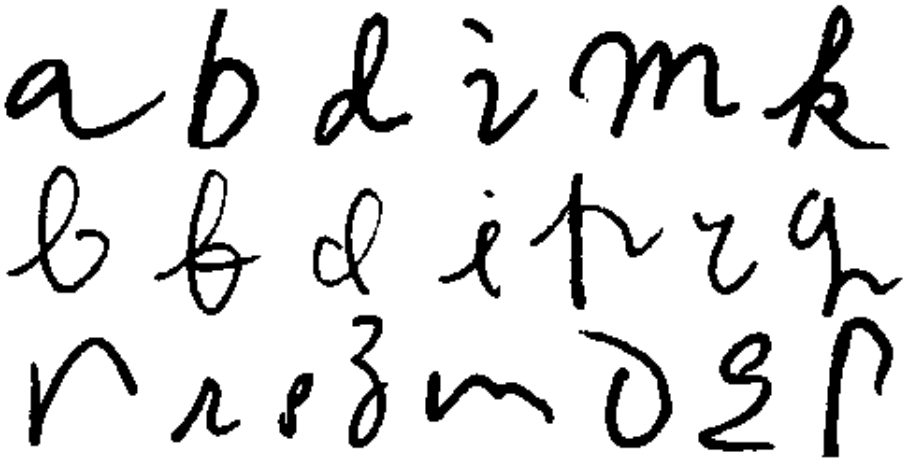


Fig. 5. Samples in CENPARMI lowercase database

Second, we investigate the effect of the number of prototype vectors on the GLVQ performance. Too many prototype vectors result in overfitting the data; too few prototype vectors can not capture the distribution of samples within each class. Fig. 7 shows the relationship between GLVQ performance and the number of prototype vectors of each class.

Finally, we verify that minimizing the MSE error does not directly result in the minimization of Bayes risk. Here the mean squared error is defined by

$$MSE = \frac{1}{N} \sum_{i=1}^{N} \min_{m_{kr} \in \Lambda} \| x_i - m_{kr} \|^2 \ . \tag{9}$$

We plot function curves of MSE, the empirical loss and training error rate with the number of iterations in Fig. 8.

In Fig. 8, MSE is monotonically increasing and the empirical loss and training error rate are monotonically decreasing. That is, the minimization of empirical loss and that of training error rate are consistent. The minimization of MSE is not necessarily related to the reduction of the training error rate. This also indicates that most clustering algorithms such as SOM that minimizes the mean square error are not suitable for classification [7].

Moreover, we also tested our method on the MNIST and CENPARMI handwritten digit database [20]. Their recognition rates are respectively 99.01% and 98.10%, two promising results.

5 Conclusion

In this work, we proposed a general local learning framework to alleviate the complexity problem of classifier design. Our method is based on a general "divide

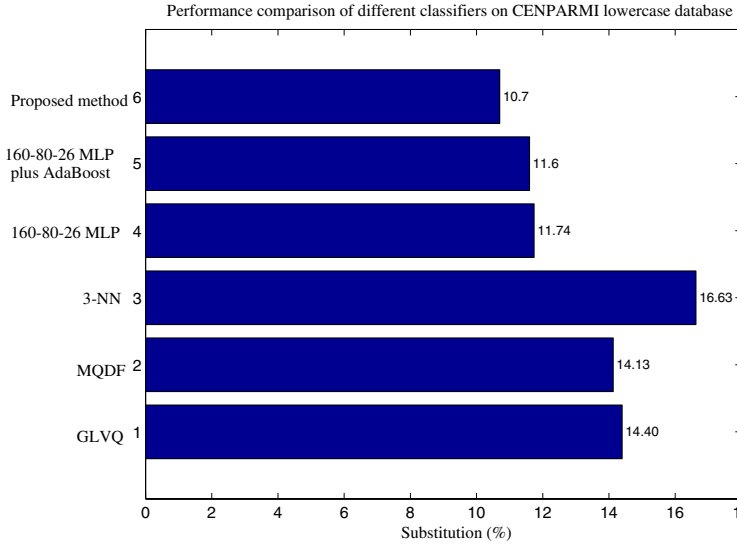


Fig. 6. Error rates of different methods on the test set of CENPARMI lowercase database, each bar represents a classifier

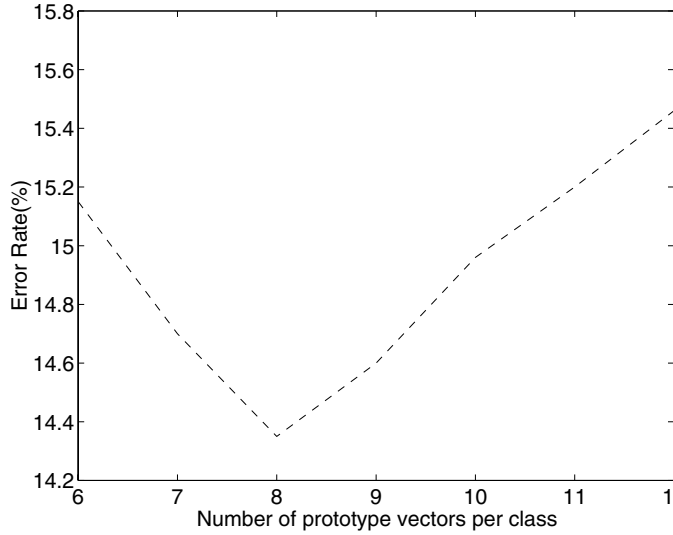


Fig. 7. Relationship between GLVQ performance and the number of reference vectors per class

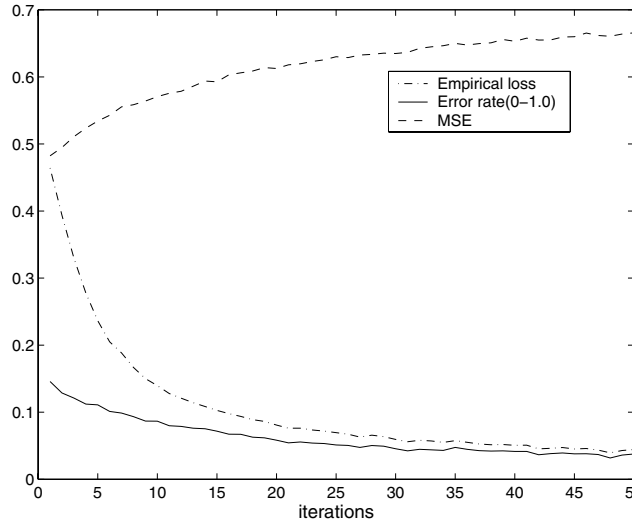


Fig. 8. Function curves of MSE, the empirical loss and training error rate versus the number of iterations

and conquer" principle and ensemble. By means of this principle, a complex real-world classification problem can be divided into many sub-problems that can be easily solved. Ensemble method is used to reduce the variance and improve a generalization ability of a neural network.

We also design an effective method to construct a good ensemble on the varied training subset. Ensemble trained on subsets can effectively capture the information of "boundary patterns" that play an important role in classification. Our method was extensively tested on several public handwritten character databases, including databases of handwritten digits and lowercase characters. It consistently achieved high performance.

The proposed method can be easily scaled to a small training set while still preserving a good performance. But it is especially suitable for a large-scale real-world classification such as Chinese and Korean character recognition and others. The results are very encouraging and strongly suggest to apply the proposed method to data mining of real world data.

References

1. Sharkey, A.J.C.: Multi-net systems. In: Sharkey, A.J.C. (eds.): Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer-Verlag, London (1999) 1–30
2. Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: Tesauro, G., Touretzky, D., Leen, T. (eds.): Advances in Neural Information Processing Systems, Vol. 7. MIT Press, Cambridge MA (1995) 231–238

3. Tumer, K., Ghosh, J.: Linear and order statistics combiners for pattern classification. In: Sharkey, A.J.C. (eds.): Combining Artificial Neural Nets: Ensemble and Modular Multi-Net Systems. Springer-Verlag, London (1999) 127–161
4. Breiman, L.: Bagging predictors. Machine Learning. **24(2)** (1996) 123–140
5. Freund, Y., Schapire, R.: Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning, Bari, Italy (1996) 148–156
6. Jocobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation. **3(1)** (1991) 79–87
7. Kohonen, T.: Self-Organizing Maps. Springer-Verlag, Berlin, Germany, 2nd Edition (1997)
8. Diamantini, C., Spalvieri, A.: Quantizing for minimum average misclassification risk. IEEE Trans. Neural Network. **9(1)** (1998) 174–182
9. Kohenon, T.: The self organizing map. Proc. IEEE. **78(9)** (1990) 1464–1480
10. Diamantini, C., Spalvieri, A.: Certain facts about kohonen's lvq1 algorithm. IEEE Trans. Circuits Syst. I. **47** (1996) 425–427
11. Juang, B.H., Katagiri, S.: Discriminative learning for minimum error classification. IEEE Trans. Signal Processing. **40(2)** (1992) 3043–3054
12. Katagiri, S., Lee, C.H., Juang, B.H.: Discriminative multilayer feedforward networks. In: Proc. IEEE Workshop Neural Network for Signal Processing. Piscataway, NJ (1991) 11–20
13. Sato, A., Yamada, K.: Generalized learning vector quantization. In: Advances in Neural Information Processing Systems. Vol. 8. MIT Press, Cambridge, MA (1996) 423–429
14. Sato, A., Yamada, K.: A formulation of learning vector quantization using a new misclassification measure. Proc. of ICPR'98 (1998) 332–325
15. Sato, A., Yamada, K.: An analysis of convergence in generalized lvq. Proc. of ICANN'98. (1998) 171–176
16. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: Proceedings of the Fifteenth International Conference on Machine Learning. Morgan Kaufmann, San Francisco, CA (1998) 91–99
17. Fujisawa, Y., Shi, M., Wakabayashi, T., Kimura, F.: Handwritten numeral recognition using gradient and curvature of gray scale image. In: Proceedings of International Conference on Document Analysis and Recognition. India (1999) 277–280
18. Srihari, S.N.: Recognition of handwritten and machine-printed text for postal address interpretations. Pattern Recognition Letters. **14** (1993) 291–302
19. Matsui, T., Tsutsumida, T., Srihari, S.N.: Combination of stroke/background structure and contour-direction features and handprinted alphanumeric recognition. In: Proc. Int. Workshop on Frontiers in Handwriting Recognition. Taipei, Taiwan, Republic of China (1994) 87–96
20. Dong, J.X., Krzyzak, A., Suen, C.Y.: A multi-net learning framework for pattern recognition. In: International Conference on Document Analysis and Recognition. accepted. Washington, USA (2001)

Mirror Image Learning for Handwritten Numeral Recognition

Meng Shi, Tetsushi Wakabayashi, Wataru Ohyama, and Fumitaka Kimura

Faculty of Engineering, Mie University, 1515 Kamihama, Tsu, 514-8507, Japan
{meng, waka, ohyama, kimura}@hi.info.mie-u.ac.jp
http://www.hi.info.mie-u.ac.jp

Abstract. This paper proposes a new corrective learning algorithm and evaluates the performance by handwritten numeral recognition test. The algorithm generates a mirror image of a pattern which belongs to one class of a pair of confusing classes and utilizes it as a learning pattern of the other class. Statistical pattern recognition techniques generally assume that the density function and the parameters of each class are only dependent on the sample of the class. The mirror image learning algorithm enlarges the learning sample of each class by mirror image patterns of other classes and enables us to achieve higher recognition accuracy with small learning sample.
Recognition accuracies of the minimum distance classifier and the projection distance method were improved from 93.17% to 98.38% and from 99.11% to 99.37% respectively in the recognition test for handwritten numeral database IPTP CD-ROM1 [1].

1 Introduction

This paper proposes a new corrective learning algorithm and evaluates the performance by handwritten numeral recognition test. The algorithm generates a mirror image of a pattern which belongs to one class of a pair of confusing classes and utilizes it as a learning pattern of the other class. The mirror image learning is a general learning method which can be widely applied to a linear classifier employing the Euclidean distance, and quadratic classifiers based on CLAFIC(Class Featuring Information Compression) [2], the subspace method [3], and the projection distance [4]. It is also applicable to the autoassociative neuralnetwork classifier [5]. The mirror image of a pattern is generated in respect to the mean vector of the linear classifier, and the minimum mean square error hyperplane of the quadratic classifiers.

Statistical pattern recognition techniques generally assume that the density function and the parameters of each class are only dependent on the sample of the class. To improve the classifier performance beyond the limitation due to the assumption several learning algorithms which exploit learning patterns of other classes (counter classes) have been proposed [6], [7], [8].

The ALSM (Averaged Learning Subspace Method) [6] adaptively modifies the basis vectors of a subspace (hyperplane) by subtracting the autocorrelation

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 239–248, 2001.

matrix for counter classes from the one of the own class. However the difference of the autocorrelation matrixes is not guaranteed to be positive semidefinite and does not has the meaning as a measure of variance. Since the constraint of the positive semidefiniteness is not preserved, the generality of the learning can be rather reduced while the extra freedom of learning is gained. The mirror image learning preserves the positive semidefiniteness of the autocorrelation matrix and the covariance matrix even if the mirror image patterns are involved in the learning sample.

The GLVQ (Generalized Learning Vector Quantization) [7] modifies the representative vectors of a pair of confusing classes so that the representative vector of the correct class approaches an input pattern and the one of the counter class goes away from the input pattern. The GLVQ algorithm is directly applied to modify the mean vectors of the minimum distance classifier and the nearest neighbor classifier, but can not be directly applied to modify and optimize the autocorrelation matrix and the covariance matrix. The mirror image learning algorithm modifies these parameters as the mirror image patterns increase in the learning sample. The mirror image learning algorithm enlarges the learning sample of each class by mirror image patterns of counter classes and enables us to achieve higher recognition accuracy with small learning sample.

2 Distance Functions for Classification

2.1 Euclidean Distance

The Euclidean distance between the input pattern and the mean vector is defined by

$$g_l^2(X) = \|X - M_l\|^2 \tag{1}$$

where X is the input feature vector of size (dimensionality) n, M_l is the mean vector of class l. The input vector is classified to such class l^* that minimizes the Euclidean distance. Hereafter the subscript l denoting the class is omitted for simplicity's sake.

2.2 Projection Distance

The projection distance [5] is defined by

$$g^2(X) = \|X - M\|^2 - \sum_{i=1}^{k} \{\Phi_i^T (X - M)\}^2 \tag{2}$$

and gives the distance from the input pattern X to the minimum mean square error hyperplane which approximates the distribution of the sample, where Φ_i denotes the i-th eigenvector of the covariance matrix, and k is the dimensionality of the hyperplane as well as the number of the dominant eigen vectors ($k < n$). When $k = 0$ the projection distance reduces to the Euclidean distance.

Fig. 1 shows an example of decision boundary in two dimensional feature space ($n = 2$, $k = 1$). This figure shows that the first principal axis approximates the distribution with minimum mean square error, and the distance from an input pattern X to the axis determines the class. It should be noted that the decision boundary in this figure consists of a pair of lines (asymptotic lines of a hyperbola) which is degenerated special case of a quadratic curve. In general the decision boundary consists of quadratic hypersurfaces.

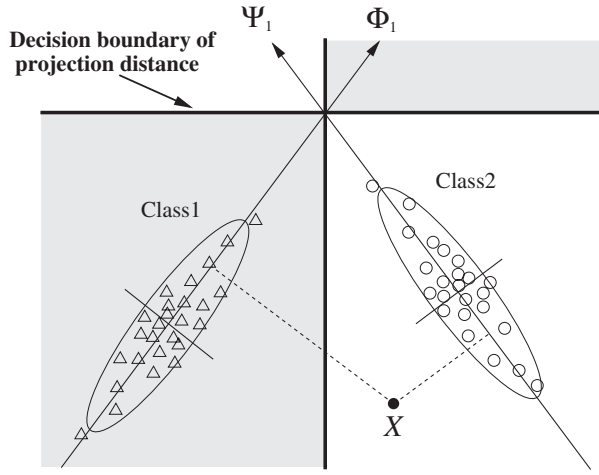


Fig. 1. An example of decision boundary of projection distance in two dimensional feature space.

2.3 Subspace Method

For a bipolar distribution on a spherical surface with $\|X\| = 1$ the mean vector M is a zero vector ($M = 0$) because the distribution is symmetric in respect to the origin [12]. Then the projection distance for the distribution is given by

$$g^2(X) = 1 - \sum_{i=1}^{k} \{\Phi_i^T X\}^2 \tag{3}$$

where Φ_i is the i-th eigenvector of the autocorrelation matrix. The second term of (3) is used as the similarity measure of CLAFIC and the subspace method.

Since the similarity measure and the Euclidean distance are reduced to a special case of the projection distance, the mirror image learning for the projection distance is described below.

3 Corrective Learning by Mirror Image

3.1 Generation and Learning of Mirror Images

When a pattern X of a class (class 1) is misclassified to a counter class (class 2), the minimum mean square error hyperplane of the class 2 is kept away from the pattern X as follows (Fig. 2).

The mirror image X' of the pattern X in respect to the hyperplane of class 2 is given by

$$X' = 2Y - X \tag{4}$$

where Y is the projection of X to the hyperplane. This mirror image X' is added to the learning sample of class 2 to keep the hyperplane away from the pattern X.

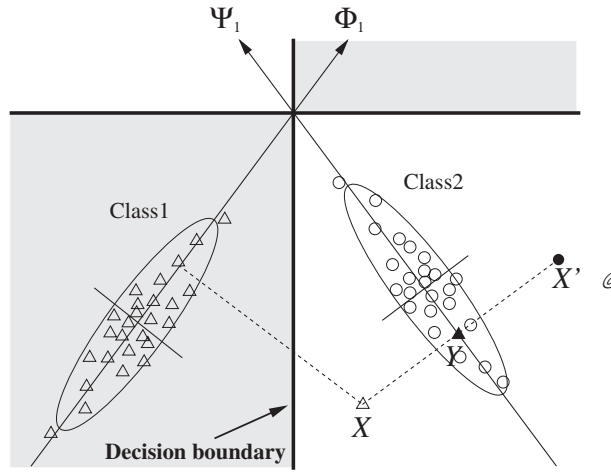


Fig. 2. Generation and learning of mirror images.

The projection Y on the hyperplane is given by a truncated KL-expansion [11],

$$Y = \sum_{i=1}^{k} \Phi_i^T (X - M)\Phi_i + M \tag{5}$$

where M, Φ_i are the mean vector and the eigenvector of the covariance matrix of class 2, respectively.

For the Euclidean distance ($k = 0$) the projection Y reduces to the mean vector M ($Y = M$), i.e. X' is the mirror image in respect to M.

For the autoassociative neuralnetwork classifier the projection Y is given by the output of the network.

While adding the mirror image X' to the learning sample of class 2, the pattern X (copy of X) is added to the sample of class 1 to move the hyperplane of class 1 toward the pattern X. As a result the decision boundary shifts to the side of class 2. After the pairs of X and X' for all learning patterns misclassified by the projection distance are added to the learning sample, the mean vector and the covariance matrix are calculated for each class, then the similar procedure is repeated until there is no change in the number of misclassifications. In the procedure, the mirror image learning is not applied recursively to those generated patterns.

3.2 Mirror Image Learning with Margin

If the number of misclassified pattern is too small the mirror image learning quickly converges close 100% correct recognition for the learning sample, before the recognition rate for the test sample is significantly improved. In order to supply the lack of misclassified patterns, confusing patterns near to the decision boundaries are extracted and utilized to generate the mirror images (Fig. 3).

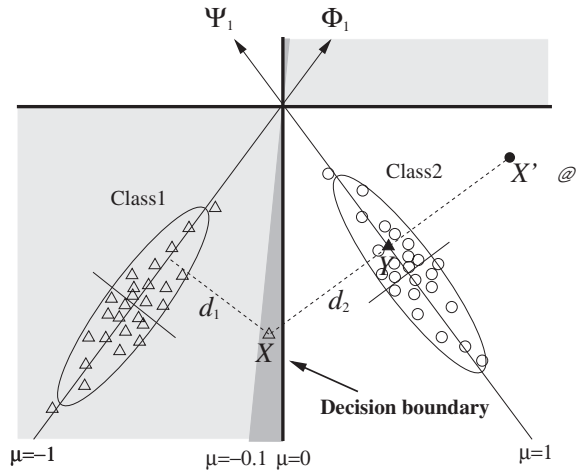


Fig. 3. Mirror image learning with margin.

A proximity measure μ of a pattern X to a decision boundary is defined by

$$\mu(X) = \frac{d_1(X) - d_2(X)}{d_1(X) + d_2(X)} \tag{6}$$

where $d_1(X)$ is the distance between X and the hyperplane of its own class, and $d_2(X)$ is the minimum distance between X and the hyperplane of the other classes. The range of μ is $[-1, 1]$, and if μ is positive (negative) the classification is wrong (correct). For a pattern X on the decision boundary $\mu = 0$, and for

the one on the hyperplane of class 1 (class 2) $\mu = -1$ ($\mu = 1$). Even if the μ is negative (correct classification) but is close to zero, the pattern is selected to generate the mirror image to enlarge the learning sample, i.e. a pattern X of class 1 is selected for mirror image generation if

$$\mu \geq \mu_t \qquad (-1 \leq \mu_t \leq 0) \tag{7}$$

for a threshold μ_t. The smaller the μ_t is, the more learning patterns are selected for the mirror image generation.

4 Performance Evaluation

4.1 Used Sample and Experiment

The handwritten ZIP code database IPTP-CDROM1 [1] provided by Institute for Posts and Telecommunications Policy is used in this experiment. The CDROM contains three digit handwritten ZIP code images collected from real Japanese new year greeting cards. The writing style and equipments have wide rage of variation. The size of image is 240 dot x 480 dot in height and width respectively, and the gray scale is 8bit (256 levels). Fig. 4 shows examples of binary images of ZIP codes. The total number of the images is 12,000 (36,000 numerals), and about a half is used for learning, and the rest for test. A series of preprocessing such as binarization and character segmentation [1] are first applied to generate binary numeral images of 120 dot x 80 dot in height and width respectively.

A feature vector of size 400 was extracted from each numeral image by the gradient analysis method [9], [10]. The mean vectors and (the eigen vectors

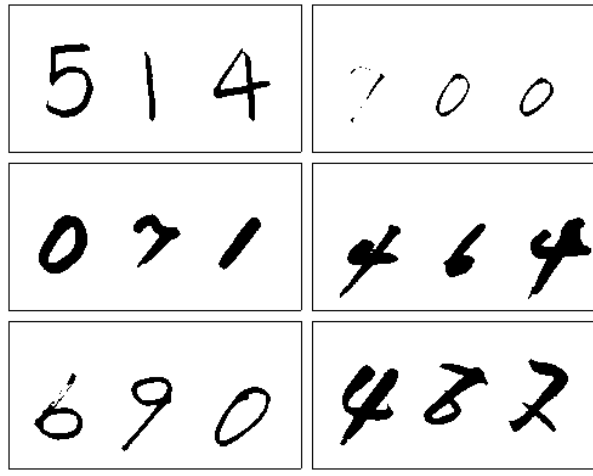


Fig. 4. Examples of binary ZIP code image.

of) the covariance matrix of the feature vectors are calculated for the learning sample. In each iteration the orth-images of the patterns for which $\mu \geq \mu_t$ are added to the sample of the correct class, and the mirror images to the sample of counter class.

4.2 Experimental Result for Euclidean Distance

Fig. 5 shows the effect of the mirror image learning for the Euclidean distance classifier. The recognition accuracy nearly converges to its maximum at about 1000 times of iteration. Table 1 shows that the recognition accuracy for the test sample is improved from 93.17% to 98.38%, while the one for the learning sample from 93.86% to 99.57%. For the Euclidean distance classifier the most significant improvement was achieved for $\mu_t = 0$. This result shows that there is no need to introduce the margin to increase the mirror image learning patterns since the Euclidean distance classifier yields enough number of misclassified patterns.

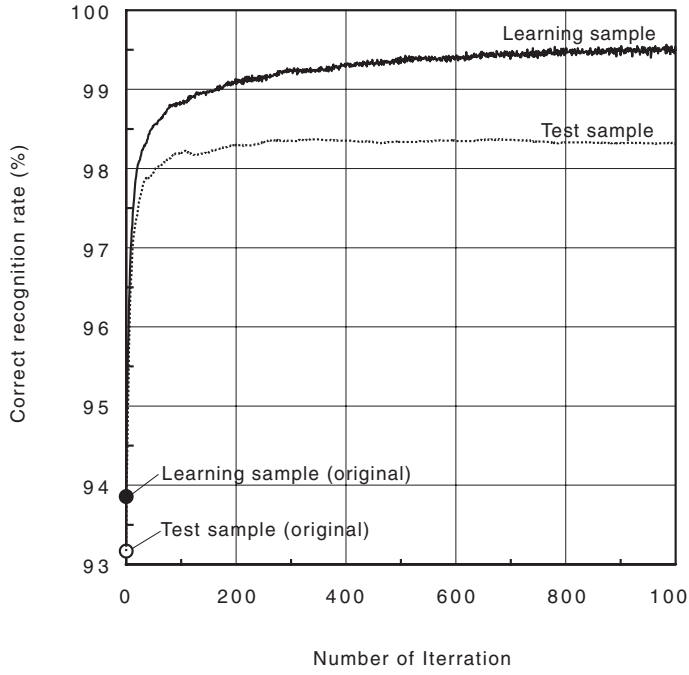


Fig. 5. Experimental results for Euclidean distance.

4.3 Experimental Result for Projection Distance

The number of used eigenvectors was fixed to 50 (k =50) and the threshold of the margin was varied as μ_t = 0, -0.2, -0.3, -0.4.

Fig. 6 shows the result of experiment for μ_t = 0, -0.3, -0.4. Without the margin (μ_t = 0) the recognition rate was hardly improved for the test sample in spite of its convergence to nearly 100% for the learning sample. However if the proper margin was introduced ($\mu_t = -0.3$) the recognition rate for the test sample was improved from 99.11% to 99.32% after 100 times of iteration. The peak rate was 99.37% (Table 1). For $\mu_t = -0.4$ the recognition rates for both of the learning sample and the test sample peaked at smaller times of iterations and were deteriorated by further iterations.

5 Conclusion

The recognition rate of the minimum distance classifier employing the Euclidean distance was improved by the mirror image learning. Since the Euclidean dis-

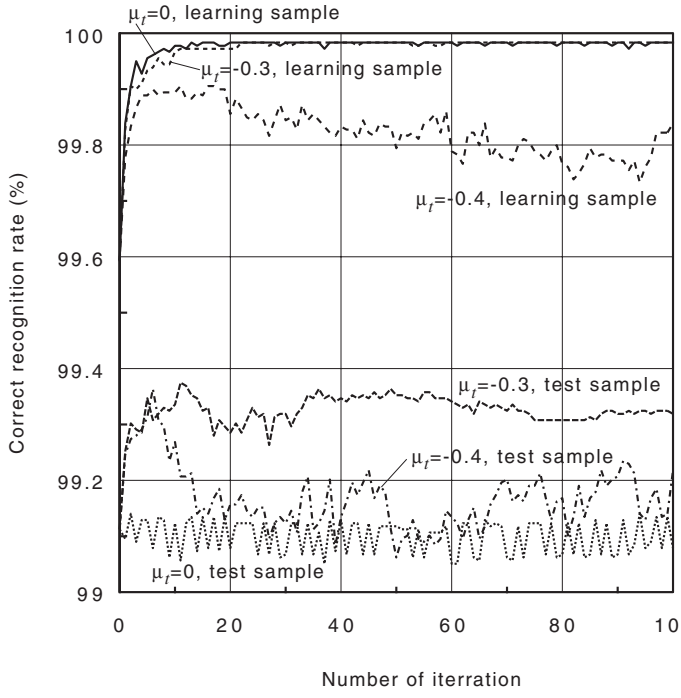


Fig. 6. Experimental results for projection distance (μ_t = 0, -0.3, -0.4).

Table 1. Result of handwritten numeral recognition by mirror image learning.

		Correct recognition rate (%)			
		Euclidean distance		Projection distance	
		Learning	Test	Learning	Test
Original method		93.86	93.17	99.59	99.11
Mirror image learning	$\mu = 0$	99.57	98.38	99.98	99.14
	$\mu = -0.2$	98.55	98.33	99.98	99.32
	$\mu = -0.3$	97.94	97.82	99.98	99.37

tance classifier yields enough number of misclassified patterns, the mirror image learning with no margin achieved the best recognition accuracy.

The recognition accuracy of the projection distance classifier was improved by the mirror image learning with the margin which extracts the confusing patterns near to the decision boundary to generate the mirror images. The effect of the mirror image learning is enhanced by the margin. Recognition rate of the projection distance classifier was improved from 99.11% to 99.37% in the recognition test for handwritten numeral database IPTP CD-ROM1.

Further studies on
(1) comparative performance evaluation with ALSM and GLVQ,
(2) effectiveness for small sample classification problems,
(3) evaluation test by Chinese character recognition,
(4) application to other than the projection distance classifier,
are remaining as future research topics.

Acknowledgement

The authors would like to acknowledge the support of Institute for Posts and Telecommunications Policy for providing us the IPTP CDROM1.

References

1. K. Osuka, T. Tsutsumida, S. Yamaguchi, K. Nagata, "IPTP Survey on Handwritten Numeral Recognition," IPTP Research and Survey Report (English translation), R-96-V-02, June 1996.
2. S. Watanabe, N. Pakvasa, "Subspace method of pattern recognition," Proc. 1st IJCPR, 1973.
3. E. Oja, "Subspace Methods of Pattern recognition," Research Studies Press, England, 1983.
4. M. Ikeda, H. Tanaka and T. Moto-oka, "Projection distance method for recognition of hand-written characters (in Japanese)," Trans. IPS Japan, vol.24, no.1, pp.106-112, 1983.

5. F. Kimura, S. Inoue, T. Wakabayashi, S. Tsuruoka, Y. Miyake, "Handwritten Numeral Recognition using Autoassociative Neural Networks," Proc. of 14th International Conference on Pattern Recognition (ICPR'98), vol.1, pp.166-171, Brisbane, Australia, 1998.
6. E. Oja, and M. Kuusela, "The ALSM algorithm - an improved subspace mothod of classification," Pattern Recognition, vol.16, no.4, pp.421-427, 1983.
7. A. Sato and K. Yamada, "Generalized learning vector quantization," Advances in Neural Information Processing 8, Proc. of the 1995 Conference, pp.423-429, MIT Press, Cambridge, MA, USA, 1996.
8. T. Fukumoto, T. Wakabayashi, F. Kimura and Y. Miyake, "Accuracy Improvement of Handwritten Character Recognition by GLVQ," Proc. of 7th International Workshop on Frontiers in handwriting recognition (IWFHR-VII), pp.271-280, Amsterdam, The Netherlands, 2000.
9. F. Kimura, T. Wakabayashi, S. Tsuruoka and Y. Miyake, "Improvement of Handwritten Japanese Character Recognition Using Weighted Direction Code Histogram," Pattern Recognition, vol.30, no.8, pp.1329-1337, 1997.
10. T. Wakabayashi, S. Tsuruoka, F. Kimura and Y. Miyake, "Increasing the feature size in handwritten numeral recognition to improve accuracy," Systems and Computers in Japan (English Edition), Scripta Technica, Inc. vol.26, no.8, pp.35-44, 1995.
11. K. Fukunaga, "Introduction to Statistical Pattern Recognition, Second Edition," Academic Press, 1990.
12. K. V. Mardia, "Statistics of Directional Data," pp.212, Academic Press, New York 1972.

Face Detection by Aggregated Bayesian Network Classifiers

Thang V. Pham, Marcel Worring, and Arnold W.M. Smeulders

ISIS, Informatics Institute, University of Amsterdam
Kruislaan 403, 1098 SJ Amsterdam, The Netherlands
{vietp, worring, smeulders}@science.uva.nl

Abstract. A face detection system is presented. A new classification method using forest-structured Bayesian networks is used. The method is used in an aggregated classifier to discriminate face from non-face patterns. The process of generating non-face patterns is integrated with the construction of the aggregated classifier using the bagging method. The face detection system performs well in comparison with other well-known methods.

1 Introduction

Face detection is an important step in any automatic face recognition system. Given an image of arbitrary size, the task is to detect the presence of any human face appearing in the image. Detection is a challenging task since human faces may appear in different scales, orientations (in-plane rotations), and with different head poses (out-of-plane rotations). The imaging conditions, including illumination direction and shadow, also affect the appearance of human faces. Moreover, human faces are non-rigid objects, as there are variations due to varying facial expressions. Presence of other devices such as glasses is another source of variation. Facial attributes such as make-up, wet skin, hairs and beards also contribute substantially to the variation of facial appearance. In addition, the appearance differences among races, and between male and female are considerable. A successful face detection system should be able to handle the multiple sources of variation.

A large number of face detection methods have been proposed in literature. Face detection methods can be broadly divided into: model-based detection, feature-based detection and appearance-based detection.

In the model-based approach, various types of facial attributes such as the eyes, the nose and the corner of the mouth are detected by a deformable geometrical model. By grouping the facial attributes based on their known geometrical relationships, faces are detected [6,16]. A drawback of this approach is the detection of facial attributes is not reliable [6], which leads to systems that are not robust against varying facial expressions and presence of other devices. This approach is better suited for facial expression recognition as opposed to face detection.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 249–262, 2001.

Among the feature-based approach, the most obvious feature is color. It is a rather surprising finding that the human skin color falls into a small range in different color spaces regardless of race [15]. Many researchers have taken advantage of this fact in their approach to the problem. Typically, regions with skin color are segmented to form face candidates. Candidates are further verified on the bases of the geometric face model. We choose not to use color information in this paper. It is partly because of the lack of a common color test set to evaluate different methods.

In the appearance-based approach, human faces are treated as a pattern directly in terms of pixel intensities [12,10]. A window of fixed size $N \times M$ is scanned over the image to find faces. The system may search for faces at multiple image scales by iteratively scaling down the image with some factor. At the core of the system is a classifier discriminating faces from non-face patterns. Each intensity in the window is one dimension in the $N \times M$ feature space. The appearance-based methods are often more robust than model-based or featured-based methods because various sources of variations can be handled by their presence in the training set.

This paper presents a face detection system in the appearance-based approach. The one class classification problem needs to be addressed because it is not possible, or meaningful, to obtain a representative set of non-face patterns for training. Furthermore, because of the manifold of sources of variation, a complex decision boundary is anticipated. In addition, the classification methods should have a very low false positive rate since the number of non-face patterns tested is normally much higher than that of face patterns. Also due to a large number of patterns which need to be tested, a fast classification step is desirable.

The paper is organized as follows. The next section gives an overview of appearance-based classification methods. The construction of an aggregated classifier is described in section 3. Section 4 presents a new classification method using forest-structured Bayesian networks. The face detection system is described in section 5. Experimental results are given in section 6.

2 Literature on Appearance-Based Face Detection

It is the classification method that characterizes different appearance-based face detection systems. Many techniques from statistical pattern recognition have been applied to distinguish between faces and non-face patterns.

Let $X = \{X_1, X_2, \ldots, X_n\}$, where $n = N \times M$, be a random variable denoting patterns spanning the $N \times M$-dimensional vector space $\mathcal{R}$. Let $x = \{x_1, x_2, \ldots, x_n\}$ be an instantiation of X. In addition, let $Y = \{0, 1\}$ be the set of class labels, face and non-face respectively. Furthermore, let the two class conditional probability distribution are $P_0(X)$ and $P_1(X)$. Once both $P_0(X)$ and $P_1(X)$ are estimated, the Bayes decision rule [3] may be used to classify a new pattern:

$$\varphi(x) = \begin{cases} 0 \text{ if } \quad \log \frac{P_0(x)}{P_1(x)} \geq \lambda \\ 1 \text{ otherwise} \end{cases} \tag{1}$$

where λ is an estimation of the log-ratio of the prior probability of the two classes. When it is not possible to obtain such approximation, one may assume equal class prior probabilities, that is $\lambda = 0$. This leads to the maximum likelihood decision rule. This leaves the question how to learn $P_y(X)$ effectively.

Moghaddam and Pentland [7] use principle component analysis to estimate the class conditional density. The vector space $\mathcal{R}$ is transformed into principle subspace E spanned by the V eigenvectors corresponding to the V largest eigenvalues and its complement $\bar{E}$ composed of the remaining eigenvectors. The authors show that in case of a Gaussian distribution, $P_y(X)$ can be approximated using the V components in the subspace E only. In case $P_y(X)$ cannot be adequately modeled using a single Gaussian, a mixture-of-Gaussians model can be used. A drawback of this method is that no guidelines are given to determine the number of dimension V. In addition, as each pattern is projected on to a subspace before classification, a matrix multiplication is involved. This is not desirable when the classification time is an important factor.

Sung and Poggio [12] present a face detection system which models $P_0(X)$ and $P_1(X)$, each by six Gaussian clusters. To classify a new pattern, a vector of distances between the pattern and the model's 12 clusters is computed, then fed into a standard multilayer perceptron network classifier. A preprocessing step is applied before classification to compensate for sources of image variation. It includes illumination gradient correction and histogram equalization. A shortcoming of this method is that there is no rule for selecting the number of Gaussian clusters.

The paper by Rowley et al. [10] is representative for a larger class of papers considering neural networks for face detection. A retinally connected neural network is used. There are three types of hidden units aiming at detecting different facial attributes that might be important for face detection. The network has a single, real-valued output. The preprocessing step in [12] is adopted. The system performs well on the CMU test set [10].

The naive Bayes classifier is used in [11]. Each pattern window is decomposed into overlapping subregions. The subregions are assumed statistically independent. Hence, $P_y(X)$ can be computed as:

$$\begin{aligned} P_y(X) &= P_y(\{R_i, P_i\}_{i=1}^{N_r}) \\ &= \prod_{i=1}^{N_r} P_y(R_i, P_i) \end{aligned} \tag{2}$$

for $y \in \{0, 1\}$. R_i is the subregion of X at location P_i and N_r is the number of subregions. The method has the power of emphasizing distinctive parts and encoding geometrical relations of a face, and hence contains elements of a model-based approach as well. A drawback of this method is the strong independence assumption. This might not lead to high classification accuracy because of the inherent dependency among overlapping subregions.

Colmenarez and Huang [2] use first order Markov processes to model the face and non-face distributions:

$$P_y(X|S) = P_y(X_{S_1}) \prod_{i=2}^{n} P_y(X_{S_i}|X_{S_{i-1}}) \tag{3}$$

for $y \in \{0, 1\}$. S is some permutation of $(1, \ldots, n)$ and used as a list of indices. The learning procedure searches for an S_m maximizing the Kullback-Leiber divergence between the two distributions $D(P_0(X)||P_1(X))$:

$$S_m = \arg\max_S D(P_0(X|S)||P_1(X|S)) \tag{4}$$

where $D(P_0(X)||P_1(X))$ is defined as:

$$D(P_0(X)||P_1(X)) = \sum_{x \in R} P_0(x) \log \frac{P_0(x)}{P_1(x)} \tag{5}$$

The Kullback-Leiber divergence is a non-negative value and equals 0 only when the two distributions are identical. The Kullback-Leiber divergence is a measure of the discriminative power between the probability distributions of the two classes [5]. By maximizing this measure, it is expected that a high classification accuracy can be achieved. The maximization problem, in this case, is equivalent to the traveling salesman problem [4]. An heuristic algorithm is applied to find an approximate solution. An advantage of this approach is that both training and classification steps are very fast.

Osuna et al. [8] apply support vector machines [14] to the face detection problem, which aims at maximizing the margin between classes. In order to train a large data set with vector support, a decomposition algorithm is proposed, in which a subset of the original data set is used. It is then updated iteratively to train the classifier.

One common characteristic of all methods is that they try to capture the decision boundary by the model supported by their classifiers. However, for classes with multiple sources of variation such as human faces, the decision boundary can be very complex. This might lead to poor accuracy performance for methods that can model simple decision boundaries. It might also lead to complex classifiers with a slow classification step. Hence, there is a need for a method which can model a complex decision boundary while allowing fast classification.

3 Data Space Exploitation and Aggregated Classifiers

In this section, we present a method which handles a complex decision boundary by using multiple classifiers in aggregation. We adopt the bagging method [1] for constructing aggregated classifiers because it allows a natural way for solving the one-class classification problem. First, we give an overview of the bagging method. We then apply it to the face detection problem.

3.1 Bagging

In [1], Breiman introduces the bagging method for generating multiple versions of a classifier to create an aggregated classifier.

A general learning set $\mathcal{L}$ consists of data $\{(y^t, x^t), t = 1, .., T\}$ where the x's are the patterns and the y's are their corresponding classes. The learning set is used to form a classifier $\varphi(x|\mathcal{L})$, that is the class of a new pattern x is determined by $\varphi(x|\mathcal{L})$. When a sequence of learning sets $\{\mathcal{L}_k; k = 1, .., K\}$, drawn from the same underlying distribution as $\mathcal{L}$ is available, one can form a sequence of classifiers $\varphi_k(x, \mathcal{L}_k)$ to make an aggregated classifier, $\varphi_A(x)$.

When y is numerical, $\varphi_A(x)$ can take the average value $\bar{\varphi}(x|\mathcal{L})$ over k. When y is a class label $c \in \{1, .., C\}$, one method of aggregating $\varphi_k(x|\mathcal{L}_k)$ is by voting. Let $N_c = \#\{k; \varphi_k(x|\mathcal{L}_k) = c\}$ and take $\varphi_A(x) = \arg\max_c N_c$.

When a single learning set $\mathcal{L}$ is available, one can take repeated bootstrap samples $\{\mathcal{L}^{(B)}\}$ from $\mathcal{L}$, and form a sequence of classifiers $\{\varphi_k(x|\mathcal{L}^{(B)})\}$. In this case, the $\{\mathcal{L}^{(B)}\}$ are drawn from $\mathcal{L}$ at random with replacement. Each sample in $\mathcal{L}$ may appear repeated times or not at all in any particular $\mathcal{L}^{(B)}$. The aggregated classifier can be formed by averaging or voting.

So far we have followed [1]. We adapt it for the one-class classification problem in the next section.

3.2 Bagging for One-Class Classification

A special case of the bagging method is used here for the face detection system. Let $\{\mathcal{L}_k; k = 1, .., K\}$ denote the K data sets to be created in order. Let φ_i denote the aggregated classifier formed by using $\{\mathcal{L}_k; k = 1, .., i\}$ for $i = 1, .., K$. The procedure for creating the data set is as follows:

1. Consider a set of face patterns $\mathcal{L}^a$. In addition, initially a set of non-face patterns $\mathcal{L}_1^{\bar{a}}$ is created by selecting randomly from a set of images containing no human faces. $\mathcal{L}^a$ and $\mathcal{L}_1^{\bar{a}}$ together form $\mathcal{L}_1$:

$$\mathcal{L}_1 = \mathcal{L}^a \cup \mathcal{L}_1^{\bar{a}} \tag{6}$$

2. For $i = 2, \ldots, K$, apply the face detection system using the aggregated classifier φ_{i-1} on a set of images containing no human faces. False positives returned form a set of non-face patterns $\mathcal{L}_i^{\bar{a}}$. Apparently, these cases are hard cases for classifier φ_{i-1}. This set $\mathcal{L}_i^{\bar{a}}$ and the training set of face patterns $\mathcal{L}^a$ form $\mathcal{L}_i$:

$$\mathcal{L}_i = \mathcal{L}^a \cup \mathcal{L}_i^{\bar{a}} \tag{7}$$

The number of classifiers K may be selected according to the desired classification accuracy. Because of our selection of learning sets, if any component classifier returns a non-face decision, the pattern is classified as non-face.

We argue that this technique is suited for the face detection problem. A complex decision boundary caused by the manifold of variation is modeled by

using multiple classifiers. Each has different level of difficulty of separating the two classes. Each component classifier need not be very complex, which could allow a fast classification step. In addition, the fact that a non-face pattern can be rejected at any level improves the classification time because of the normally large number of non-face patterns. The one-class problem is overcome by boostrapping of false positives. Significantly, since the same face patterns, $\mathcal{L}^a$, are used for training, the true positive rate does not degrade multiplicatively as the number of component classifiers increases. Also, because the non-face patterns are generated in a bootstrap fashion, it is expected that the false positive rate decreases multiplicatively. This allows a very low false positive rate.

4 Forest-Structured Bayesian Network Classifier

In this section a new classification method for the two-class problem is described. The method is in the same spirit as the Markov process-based method in [2]. However, forest-structured Bayesian networks are used to model the joint probability distribution of each class instead of Markov processes. We use this method in an aggregated classifier because it has a fast classification step.

Bayesian network is an efficient tool to model the joint distribution of variables [9]. The joint distribution $P_y(X_1, .., X_n)$ can be expressed using a forest structured Bayesian network as follows:

$$P_y(x) = \prod_{i=1}^{n} P_y(X_i = x_i | \Pi_i = \pi_i) \tag{8}$$

for $y \in \{0, 1\}$. Π_i denote the parent of X_i in the network structure. $P_y(X_i = x_i | \Pi_i = \pi_i)$ are estimated from the training data $\mathcal{L}_i$ (eq. (6) or (7)). Figure 1 illustrates a forest structured Bayesian network modeling the joint distribution of six random variables $\{X_1, .., X_6\}$.

We search for a network structure that maximizes the Kullback-Leiber divergence eq. (5) between the two joint distributions.

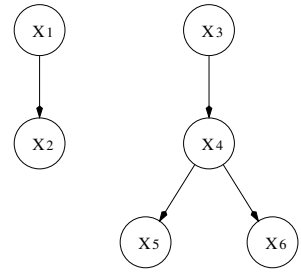


Fig. 1. A sample dependency model of six random variables with a forest structured Bayesian network:
$P(X_1, .., X_6) = P(X_1)P(X_2|X_1)P(X_3)P(X_4|X_3)P(X_5|X_4)P(X_6|X_4)$

The Kullback-Leiber divergence between two distributions in eq. (8) can be obtained as:

$$\begin{aligned} D(P_0(X)||P_1(X)) &= \sum_x P_0(x) \log \prod_{i=1}^{n} \frac{P_0(x_i|\pi_i)}{P_1(x_i|\pi_i)} \\ &= \sum_{i=1}^{n} \sum_x P_0(x) \log \frac{P_0(x_i|\pi_i)}{P_1(x_i|\pi_i)} \\ &= \sum_{i=1}^{n} \sum_{x_i} \sum_{pa_i} P_0(x_i, \pi_i) \log \frac{P_0(x_i|\pi_i)}{P_1(x_i|\pi_i)} \end{aligned} \tag{9}$$

We show that the problem of maximizing eq. (9) is equivalent to the maximum branching problem [13]. In the maximum branching problem, a branching B of a directed graph G is a set of arcs such that:

1. if (x_1, y_1) and (x_2, y_2) are distinct arcs of B then $y_1 \neq y_2$.
2. B does not contain a cycle.

Given a real value $c(v, w)$ defined for each arc of G, a maximum branching of G is a branching such that $\sum_{(v,w)\in B} c(v, w)$ is maximum. It can be seen that maximizing $D(P_0(X)\|P_1(X))$ is equivalent to finding a maximum branching of a weighted directed graph constructed from the complete graph with node x_i's plus a node x_0 with an arc from x_0 to all other nodes. $W(i, j) = \sum_{x_i} \sum_{x_j} P_0(x_i, x_j) \log \frac{P_0(x_i|x_j)}{P_1(x_i|x_j)}$ is the weight associated with each arc in the graph. There are algorithms for solving the maximum branching problem in low order polynomial time [13].

To classify a pattern x, the Bayes decision rule eq. (1) is used. Similar to the method in [2], fast classification of a pattern can be achieved by constructing a table for all possible values of a variable and its parent. By using eq. (8), the log likelihood value in eq. (1) becomes:

$$\begin{aligned} \log \frac{P_0(x)}{P_1(x)} &= \log \frac{\prod_{i=1}^{n} P_0(x_i|\pi_i)}{\prod_{i=1}^{n} P_1(x_i|\pi_i)} \\ &= \sum_{i=1}^{n} \log \frac{P_0(x_i|\pi_i)}{P_1(x_i|\pi_i)} \end{aligned} \tag{10}$$

Once all possible values of $\log \frac{P_0(X_i|\Pi_i)}{P_1(X_i|\Pi_i)}$ for all i are computed, the classification of a new pattern can be carried out with only n additions. This allows a very fast classification step.

5 Face Detection System

The architecture of the system is adopted from [10]. A window of size 20×20 is scanned over each image location to find face patterns. The size 20×20 is selected

because it is large enough to capture details of human faces, while allowing a reasonable classification time. The system searches the input image at multiple scales by iteratively scaling down the image with a scale factor of 1.2 until the image size is less than the window size.

Sources of variation are captured in the training set: illumination and shadows, facial expressions, glasses, make-up, hairs, beards, races and sexes. Limited orientation and head pose, namely frontal faces and near-frontal faces, are present.

We adopt two preprocessing operations from [12]: illumination gradient correction and histogram equalization. The former reduces the effect of heavy shadows and the latter normalizes the illumination contrast of the image. Finally, each pattern is discretized to six levels of gray values to enable the estimation of the discrete probabilities.

An aggregated classifier consisting of three Bayesian network classifiers, i.e. φ_3, is used to classify faces and non-face patterns. The number 3 was selected based on the tradeoff between the false positive rate and true positive rate (see figure 5). For $K > 3$, the true positive rate is low for the detection task.

A postprocessing step is carried out to eliminate overlapping detections. When overlapping occurs, a straightforward approach would be to select the window having the largest log likelihood value. This generates sparse maxima, of which most are false positives as is observed in [10], that is most faces are detected are detected at multiple positions nearby in place or in scale. We have repeated the experiment and arrived at the same conclusion. For each detected location, if the number of detections within a predefined neighborhood is less than a threshold, the location is rejected.

5.1 Data for Training

For the purpose of this paper, a set of 1112 face examples was gathered from the Internet without selection. Color images were converted to gray-scale images. Figure 2 gives 30 randomly selected face examples. The dataset is split into two subsets at random: 1000 faces examples are used to create the training set and 112 used to create the test set. Thirty face patterns of size 20×20 are extracted from each original face examples by rotating the images about their center points by one random less than 10 degree, scaling by one random value selected from the interval 0.9 and 1.1, translating by one random value less than 0.5 pixel, and mirroring as in [10]. Figure 3 illustrates 30 face patterns generated from one face example. In total, 33360 face patterns were created.

A set of 929 images containing no faces was also collected from the Internet. 360000 non-face patterns are extracted from the images by randomly selecting a square from an image and subsampling it to patterns of size 20×20. Figure 4 contains 30 non-face patterns. From the next level downwards, non-face patterns were generated as described in section 3.2.

The dataset of 33360 face and 360000 non-face patterns is split into two subsets at random: the training set consists of 30000 face and 160000 non-face patterns, and the test set consists of 3360 face and 200000 non-face patterns.

Fig. 2. 30 of 1112 randomly selected face examples

Fig. 3. An example of all 30 face patterns generated from each face example, yielding 30000 patterns to train the system

This test set is referred to as the pattern test set, $\mathcal{L}^T$. The face patterns of the two subsets were generated from two separate sets of face examples.

6 Experimental Results

6.1 Experiment with the Number of Component Classifiers K

Figure 5 shows the receiver operating characteristic curves for the four aggregated classifiers φ_1, φ_2, φ_3 and φ_4 on the pattern test set $\mathcal{L}^T$. At a low false positive rate an aggregated classifier with higher value of K achieves higher true positive rate.

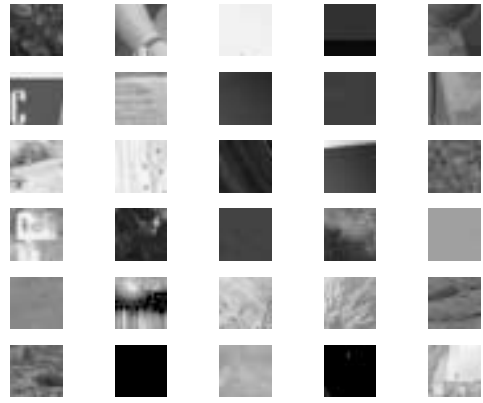


Fig. 4. 30 non-face patterns randomly selected from the set of 160000 non-face patterns

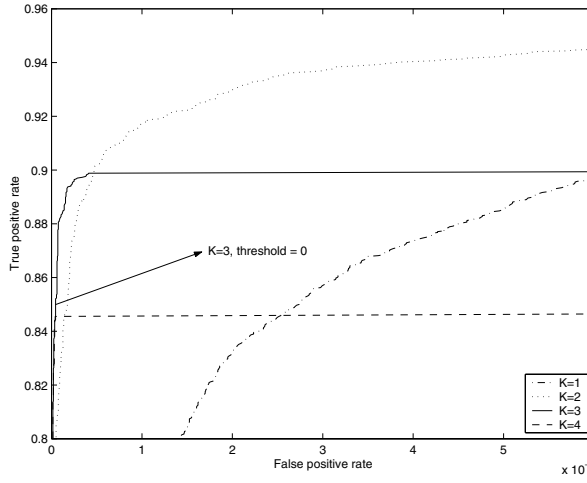


Fig. 5. The Receiver Operating Characteristic (ROC) curves for φ_1, φ_2, φ_3 and φ_4 on the pattern test set $\mathcal{L}^T$

6.2 Experiment with the Bayesian Network Classifier

Figure 6 shows the Receiver Operating Characteristic (ROC) curves of the three different classifiers on the pattern test set $\mathcal{L}^T$: the Markov process classifier [2], the naive Bayes classifier [3] and our method, the Bayesian network classifier φ_1. Our method outperforms both the Markov process classifier and the naive Bayes classifier.

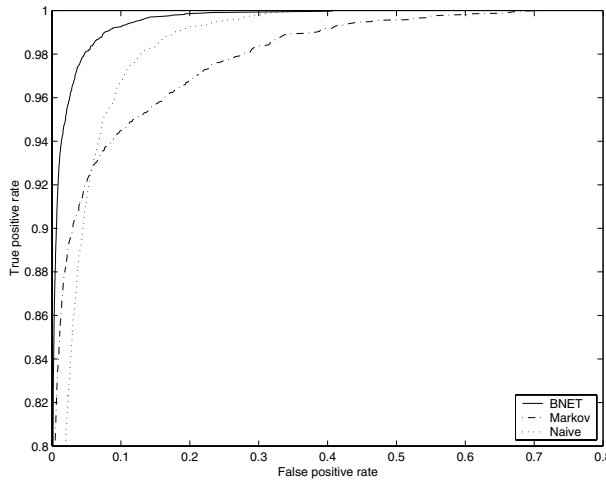


Fig. 6. The Receiver Operating Characteristic (ROC) curves of three classifiers:the Markov process based classifier [2], the naive Bayes classifier [3] and the Bayesian network classifier φ_1

As an aside, it is interesting to see that the Markov classifier performs better than the naive Bayes classifier only when the positive error rate is smaller than 6%.

6.3 Experiment on a Full Image Test Set

The system is evaluated using the CMU test set [10]. This test set consists of 130 images with a total of 507 frontal faces, including images of the MIT test set [12]. The images were collected from the World Wide Web, scanned from photographs and newspaper pictures, and digitized from broadcast television. There is a wide range of variation in image quality. It should be noted that some authors report their results on a test set excluding 5 images of line draw faces [11], which leaves this test set with 125 images with 483 labeled faces only. We use the groundtruth with 507 faces as in [10].

Table 1 shows the performance of our face detection system in comparison with systems in [10] on the CMU test sets. It can be seen that with an equivalent detection rate, Bayesian network based method gives about half the number of false detections in comparison with the neural network method [10]. Figure 7 illustrates the detection result on some images of the CMU test set.

7 Discussion and Conclusion

In this paper we have considered the face detection task as a representative of the one class classification problem where the class is subjected to many

Table 1. Evaluation of the performance of the aggregated Bayesian networks, BN, as compared to the neural network, NN [10] on the CMU test set [10]. The criteria are: the number of missed faces (MFs), the true detection rate (Rate) and the number of false detects (FDs)

Our system	MFs	Rate	FDs
BN	47	90.7%	264

System in [10]	MFs	Rate	FDs
NN, System 5	48	90.5%	570
NN, System 6	42	91.7%	506
NN, System 7	49	90.3%	440
NN, System 8	42	91.7%	484

sources of variation. The sources of variation include position of the face relative to the camera, illumination condition, non-rigid characteristic of the face, and presence of other devices. The appearance variation is also caused by facial attributes, differences among races, and between male and female. In addition, the classification method must have a very low false positive rate and a fast classification step.

Our face detection system performs well. On the CMU test set it achieves a detection rate of about 90%, with an acceptable number of false alarms. In comparison with other methods, our classification method using Bayesian networks outperforms related methods (namely the Markov process method [2] and the naive Bayes classifier [3], as shown in Figure 6). On the CMU test set, our system performs better than the neural network method [10]. Our system gives about half the number of false alarms at an equivalent detection rate (see Table 1).

Approximately half of the missed detections are caused by rotated angles (see Figure 7, image D). Large in-plane rotation or out-of-plane rotation are not handled with this method. When the subject has the intention of looking into the camera, false negatives are rare. In fact, the missed detection in image D is one of the very few cases. Poor image conditions, such as low brightness and strong shadows, account for about one third of the missed detections (see the three examples in image E). In order to resolve this a special image enhancement preprocessing step might help. The remaining missed detections are caused by various reasons including the sizes of the faces being too small. Among the false positives, in 30 cases out of 264, the patches do appear as human faces (see the false alarm in image E and the top two false alarms in image F). Other cases might be eliminated by further postprocessing. Given the large number of tested windows [10], our method makes only one incorrect classification out of each 300000 tests.

Because our method uses a memory-based histogram for probability density estimation, there is a limitation on the number of discrete levels to be used. During the training process, at 6 discrete levels, each histogram takes up 44 Megabytes of memory. At 8 discrete levels, each histogram would take up about 78 Megabytes. Discretization causes loss of information, but does not necessarily

Fig. 7. Output of the system on some images of the CMU test set [10]. **MFs** is the number of missed faces and **FDs** is the number of false detections

reduce the classification accuracy. With higher number of discrete levels, more training data are needed to characterize the distributions. Furthermore, we still can distinguish face patterns from non-face patterns at 6 discrete gray values. An

experiment with 4 discrete levels (data not shown) indicates a slightly degraded performance. For the purpose of this paper, 6-level discretization is appropriate.

Our system makes use of the symmetry property of the human face only implicitly by the mirroring operation on the training face examples. It is interesting to investigate how symmetry can be encoded in the Bayesian network prior to the learning phase. It is important to note, however, that structural biases and lighting may affect the symmetry property.

In conclusion, this paper presents a face detection system using an aggregation of Bayesian network classifiers. The use of an aggregated classifier is well suited for the one-class classification problem in the visual domain, where a complex decision boundary is anticipated due to many sources of variation. In addition, aggregated classifiers allow a very low false positive rate and fast detection.

References

1. L. Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996.
2. A. Colmenarez and T. Huang. Face detection with information-based maximum discrimination. In *Proc. of CVPR'97*, pages 782–787, 1997.
3. R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, Inc., 1973.
4. M. Gondran and M. Minoux. *Graphs and Algorithms*. John Wiley & Sons, 1984.
5. S. Kullback. *Information Theory and Statistics*. John Wiley, 1959.
6. T.K. Leung, M.C. Burl, and P. Perona. Finding faces in cluttered scenes using random labeled graph matching. In *Proc. of The Fifth International Conference on Computer Vision*, pages 637–644, 1995.
7. B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE PAMI*, 19(7):696–710, 1997.
8. E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. of CVPR'97*, pages 130–136, Puerto Rico, 1997.
9. J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, San Mateo, CA, 1988.
10. H. A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE PAMI*, 20(1):23–38, 1998.
11. H. Schneiderman and K. Kanade. A statistical method for 3D object detection applied to faces and cars. In *Proc. of CVPR 2000*, pages 746–751, 2000.
12. K. K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE PAMI*, 20(1):39–51, 1998.
13. R. Tarjan. Finding optimum branchings. *Networks*, 7:25–35, 1977.
14. V. N. Vapnik. *Statistical Learning Theory*. John Wiley & Sons, Inc, 1998.
15. J. Yang and A. Waibel. A real-time face tracker. In *Proc. of WACV'96*, pages 142–147, 1996.
16. K.C. Yow and R. Cipolla. Scale and orientation invariance in human face detection. In *Proceedings 7th British Machine Vision Conference*, pages 745–754, 1996.

Towards Self-Exploring Discriminating Features

Ying Wu and Thomas S. Huang

Beckman Institute
University of Illinois at Urbana-Champaign
405 N. Mathews, Urbana, IL 61801
{yingwu,huang}@ifp.uiuc.edu
http://www.ifp.uiuc.edu/~yingwu

Abstract. Many visual learning tasks are usually confronted by some common difficulties. One of them is the lack of supervised information, due to the fact that labeling could be tedious, expensive or even impossible. Such scenario makes it challenging to learn object concepts from images. This problem could be alleviated by taking a hybrid of labeled and unlabeled training data for learning. Since the unlabeled data characterize the joint probability across different features, they could be used to boost weak classifiers by exploring discriminating features in a self-supervised fashion. Discriminant-EM (D-EM) attacks such problems by integrating discriminant analysis with the EM framework. Both linear and nonlinear methods are investigated in this paper. Based on kernel multiple discriminant analysis (KMDA), the nonlinear D-EM provides better ability to simplify the probabilistic structures of data distributions in a discrimination space. We also propose a novel data-sampling scheme for efficient learning of kernel discriminants. Our experimental results show that D-EM outperforms a variety of supervised and semi-supervised learning algorithms for many visual learning tasks, such as content-based image retrieval and invariant object recognition.

1 Introduction

Characterizing objects or concepts from images is one of the fundamental research topics of computer vision. Since there could be large variations in the image appearances due to various illumination conditions, viewing directions, variations in a general concept, this task is challenging because finding effective and explicit representations is generally a difficult problem. To approach this problem, machine learning techniques could be employed to model the variations in image appearances by learning the representations from a set of training data.

For example, invariant 3D object recognition is to recognize objects from different view directions. 3D object reconstruction suggests a way to invariantly characterize objects. Alternatively, objects could also be represented by their visual appearance without explicit reconstruction. However, representing objects in the image space is formidable, since the dimensionality of the image space is intractable. Dimension reduction could be achieved by identifying invariant

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 263–277, 2001.

image features. In some cases, domain knowledge could be exploited to extract image features from visual inputs, however, many other cases need to *learn* such features from a set of examples when image features are difficult to define. Many successful examples of learning approaches in the area of face and gesture recognition can be found in the literature [4,2].

Generally, representing objects from examples requires huge training data sets, because input dimensionality is large and the variations that object classes undergo are significant. Although unsupervised or clustering schemes have been proposed [1,20], it is difficult for pure unsupervised approaches to achieve accurate classification without supervision. Labels or supervised information of training samples are needed for recognition tasks. The generalization abilities of many current methods largely depend on training data sets. In general, good generalization requires large and representative labeled training data sets.

Unfortunately, collecting labeled data can be a tedious process. In some other cases, the situations are even worse, since it maybe impossible to label all the data. Content-based image retrieval is one of such examples.

The task of image retrieval is to find as many as possible "similar" images to the query images in a given database. Early research of image retrieval is searching by manually annotating every image in a database. To avoid manual annotating, an alternative approach is content-based image retrieval (CBIR), by which images would be indexed by their visual contents such as color, texture, shape, etc. Many research efforts have been made to extract these low-level image features [8,15], evaluate distance metrics [13,16], and look for efficient searching schemes [18]. However, it is generally impossible to find a fixed distance or similarity metrics. Such task could be cast as a classification problem, i.e., the retrieval system acts as a classifier to divide the images in the database into two classes, either relevant or irrelevant [22]. Unfortunately, one of the difficulties for learning is that only very limited number of query images could be used as labeled data, so that pure supervised learning with such limited training data can only give very weak classifiers.

We could consider the integration of pure supervised and unsupervised learning by taking hybrid data sets. The issue of combining unlabeled data in supervised learning begins to receive more and more research efforts recently and the research of this problem is still in its infancy. Without assuming parametric probabilistic models, several methods are based on the SVM [6,3,7]. However, when the size of unlabeled data becomes very large, these methods need formidable computational resources for mathematical programming. Some other alternative methods try to fit this problem into the EM framework and employ parametric models [22,23], and have some applications in text classification [7,11,12]. Although EM offers a systematic approach to this problem, these methods largely depend on the *a priori* knowledge about the probabilistic structure of data distribution.

Since the labels of unlabeled data can be treated as missing values, The Expectation-Maximization (EM) approach can be applied to this problem. We assume that the hybrid data set is drawn from a mixture density distribution

of C components $\{c_j, j = 1, \ldots, C\}$, which are parameterized by $\mathbf{\Theta} = \{\theta_j, j = 1, \ldots, C\}$. The mixture model can be represented as:

$$p(\mathbf{x}|\mathbf{\Theta}) = \sum_{j=1}^{C} p(\mathbf{x}|c_j; \theta_j) p(c_j|\theta_j) \tag{1}$$

where $\mathbf{x}$ is a sample drawn from the hybrid data set $\mathcal{D} = \mathcal{L} \bigcup \mathcal{U}$. We make another assumption that each component in the mixture density corresponds to one class, i.e. $\{y_j = c_j, j = 1, \ldots, C\}$. Then, the joint probability density of the hybrid data set can be written as:

$$p(\mathcal{D}|\mathbf{\Theta}) = \prod_{\mathbf{x}_i \in \mathcal{U}} \sum_{j=1}^{C} p(c_j|\mathbf{\Theta}) p(\mathbf{x}_i|c_j; \mathbf{\Theta}) \bullet \prod_{\mathbf{x}_i \in \mathcal{L}} p(y_i = c_i|\mathbf{\Theta}) p(\mathbf{x}_i|y_i = c_i; \mathbf{\Theta})$$

The parameters $\mathbf{\Theta}$ can be estimated by maximizing *a posteriori* probability $p(\mathbf{\Theta}|\mathcal{D})$. Equivalently, this can be done by maximizing $\lg(p(\mathbf{\Theta}|\mathcal{D}))$. Let $l(\mathbf{\Theta}|\mathcal{D}) = \lg(p(\mathbf{\Theta})p(\mathcal{D}|\mathbf{\Theta}))$. A binary indicator $\mathbf{z}_i$ is introduced, $\mathbf{z}_i = (z_{i1}, \ldots, z_{iC})$. And $z_{ij} = 1$ iff $y_i = c_j$, and $z_{ij} = 0$ otherwise, so that

$$l(\mathbf{\Theta}|\mathcal{D}, \mathcal{Z}) = \lg(p(\mathbf{\Theta})) + \sum_{\mathbf{x}_i \in \mathcal{D}} \sum_{j=1}^{C} z_{ij} \lg(p(O_j|\mathbf{\Theta}) p(\mathbf{x}_i|O_j; \mathbf{\Theta})) \tag{2}$$

The EM algorithm can be used to estimate the parameters $\mathbf{\Theta}$ by an iterative hill climbing procedure, which alternatively calculates $E(\mathcal{Z})$, the expected values of all unlabeled data, and estimates the parameters $\mathbf{\Theta}$ given $E(\mathcal{Z})$. The EM algorithm generally reaches a local maximum of $l(\mathbf{\Theta}|\mathcal{D})$. It consists of two iterative steps:

- **E-step**: set $\hat{\mathcal{Z}}^{(k+1)} = E[\mathcal{Z}|\mathcal{D}; \hat{\Theta}^{(k)}]$
- **M-step**: set $\hat{\Theta}^{(k+1)} = \arg\max_{\theta} p(\Theta|\mathcal{D}; \hat{\mathcal{Z}}^{(k+1)})$

where $\hat{\mathcal{Z}}^{(k)}$ and $\hat{\Theta}^{(k)}$ denote the estimation for $\mathcal{Z}$ and Θ at the k-th iteration respectively. When the size of the labeled set is small, EM basically performs an unsupervised learning, except that labeled data are used to identify the components. If the probabilistic structure, such as the number of components in mixture models, is known, EM could estimate true parameters of the probabilistic model. Otherwise, the performance can be very bad. Generally, when we do not have such *a prior* knowledge about the data distribution, a Gaussian distribution is always assumed to represent a class. However, this assumption is often invalid in practice, which is partly the reason that unlabeled data hurt the classifier.

To alleviate such difficulties for the EM-based approaches, this paper proposes a novel approach, the *Discriminant-EM (D-EM)* algorithm, by inserting a step of discriminant analysis step into the EM iterations. Both linear and nonlinear discriminant analysis will be discussed in this paper. The proposed nonlinear method is based on kernel machines. A novel algorithm is presented for sampling

training data for efficient learning of nonlinear kernel discriminants. We did standard benchmark testing of the kernel discriminant analysis. Our experiments of the D-EM algorithm include view-independent hand posture recognition and transductive content-based image retrieval.

2 Discriminant-EM Algorithm

As an extension to Expectation-Maximization, *Discriminant-EM (D-EM)* is a self-supervised learning algorithm for such purposes by taking a small set of labeled data with a large set of unlabeled data. The D-EM algorithm loops between an expectation step, a discrimination step, and a maximization step. D-EM estimates the parameters of a generative model in a discrimination space.

The basic idea of this algorithm is to learn discriminating features and the classifier simultaneously by inserting a multi-class linear discrminant step in the standard expectation-maximization iteration loop. The basic idea of D-EM is to identify some "similar" samples in the unlabeled data set to enlarge the labeled data set so that supervised techniques are made possible in such an enlarged labeled set.

- **E-step**: set $\hat{\mathcal{Z}}^{(k+1)} = E[\mathcal{Z}|\mathcal{D}; \hat{\Theta}^{(k)}]$
- **D-step**: find a discriminant space and project data onto it
- **M-step**: set $\hat{\Theta}^{(k+1)} = \arg\max_\theta p(\Theta|\mathcal{D}; \hat{\mathcal{Z}}^{(k+1)})$

The E-step gives unlabeled data probabilistic labels, which are then used by the D-step to separate the data. D-EM makes assumption that the probabilistic structure of data distribution in the lower dimensional discrimination space is simplified and could be captured by lower order Gaussian mixtures. In this sense, the discriminant projection is not arbitrary. We will have a detailed discussion on the D-step in the next two sections, and concentrate on nonlinear discriminant analysis approaches.

D-EM begins with a weak classifier learned from the labeled set. Certainly, we do not expect much from this weak classifier. However, for each unlabeled sample $\mathbf{x}_j$, the classification confidence $\mathbf{w}_j = \{w_{jk}, k = 1, \ldots, C\}$ can be given based on the probabilistic label $\mathbf{l}_j = \{l_{jk}, k = 1, \ldots, C\}$ assigned by this weak classifier.

$$l_{jk} = \frac{p(\phi(\mathbf{x}_j)|c_k)p(c_k)}{\sum_{k=1}^{C} p(\phi(\mathbf{x}_j)|c_k)p(c_k)} \tag{3}$$

$$w_{jk} = -\lg(p(\phi(\mathbf{x}_j)|c_k)),\ k = 1, \ldots, C \tag{4}$$

Euqation(4) is just a heuristic to weight unlabeled data $\mathbf{x}_j \in \mathcal{U}$, although there may be many other choices.

After that, multiple discriminant analysis is performed on the new weighted data set,

$$\mathcal{D}' = \mathcal{L} \bigcup \{\mathbf{x}_j, \mathbf{l}_j, \mathbf{w}_j : \forall \mathbf{x}_j \in \mathcal{U}\},$$

by which the data set $\mathcal{D}'$ is projected to a new space of dimension $C-1$ but unchanging the labels and weights, i.e.,

$$\hat{\mathcal{D}} = \{\phi(\mathbf{x})_j, y_j : \forall \mathbf{x}_j \in \mathcal{L}\} \bigcup \{\phi(\mathbf{x})_j, \mathbf{l}_j, \mathbf{w}_j : \forall \mathbf{x}_j \in \mathcal{U}\}. \tag{5}$$

Then parameters $\mathbf{\Theta}$ of the probabilistic models are estimated by maximizing a posteriori probability on $\hat{\mathcal{D}}$, so that the probabilistic labels are given by the Bayesian classifier according to Equation(3). The D-EM algorithm iterates over these three steps, "Expectation-Discrimination-Maximization".

3 Linear Multiple Discriminant Analysis

Multiple discriminant analysis (MDA) is a natural generalization of Fisher's linear discriminant analysis (LDA) for the case of multiple classes [5]. The goal of MDA is to find a linear projection $\mathbf{W}$ that maps the original d_1-dimensional data space $\mathcal{X}$ to a d_2-dimensional discrimination space Δ ($d_2 \leq c-1$, c is the number of classes) such that the classes are linearly separable.

More specifically, MDA finds the best linear projection of labeled data, $\mathbf{x} \in \mathcal{X}$, such that the ratio of between-class scatter, S_B, to within-class scatter, S_W, is maximized. Let n be the size of training data set, and n_j be the size of the data set for class j. Then,

$$\mathbf{V}_{opt} = \arg\max_{\mathbf{V}} \frac{|\mathbf{V}^T S_B \mathbf{V}|}{|\mathbf{V}^T S_W \mathbf{V}|} \tag{6}$$

$$S_B = \sum_{j=1}^{c} n_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T, \tag{7}$$

$$S_W = \sum_{j=1}^{c} \sum_{k=1}^{n_j} (\mathbf{x}_k - \mathbf{m}_j)(\mathbf{x}_k - \mathbf{m}_j)^T, \tag{8}$$

where the total mean and class means are given by

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{x}_k,$$

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} \mathbf{x}_k, \ \forall j \in \{1, \ldots, c\}$$

and $\mathbf{V}_{opt} = [\mathbf{v}_1, \ldots, \mathbf{v}_{c-1}]$ will contain in its columns $c-1$ eigenvectors corresponding to $c-1$ eigenvalues, i.e.,

$$S_B \mathbf{v}_i = \lambda_i S_W \mathbf{v}_i.$$

4 Nonlinear Discriminant Analysis

Nonlinear discriminant analysis could be achieved by transforming the original data space $\mathcal{X}$ to a nonlinear feature space $\mathcal{F}$ and then performing LDA in $\mathcal{F}$. This section presents a kernel-based approach.

4.1 Kernel Discriminant Analysis

In *nonlinear* discriminant analysis, we seek a prior transformation of the data, $\mathbf{y} = \phi(\mathbf{x})$, that maps the original data space $\mathcal{X}$, to a feature space (F-space) $\mathcal{F}$, in which MDA can be then performed. Thus, we have

$$\mathbf{V}_{opt} = \arg\max_{\mathbf{V}} \frac{|\mathbf{V}^T S_B^\phi \mathbf{V}|}{|\mathbf{V}^T S_W^\phi \mathbf{V}|}, \tag{9}$$

$$S_B^\phi = \sum_{j=1}^{c} n_j (\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^T, \tag{10}$$

$$S_W^\phi = \sum_{j=1}^{c} \sum_{k=1}^{n_j} (\phi(\mathbf{x}_k) - \mathbf{m}_j)(\phi(\mathbf{x}_k) - \mathbf{m}_j)^T, \tag{11}$$

with

$$\mathbf{m} = \frac{1}{n} \sum_{k=1}^{n} \phi(\mathbf{x}_k),$$

$$\mathbf{m}_j = \frac{1}{n_j} \sum_{k=1}^{n_j} \phi(\mathbf{x}_k), \ \forall j \in \{1, \ldots, c\}.$$

In general, because we choose $\phi(\cdot)$ to facilitate *linear* discriminant analysis in the feature space $\mathcal{F}$, the dimension of the feature space may be arbitrarily large, even infinite. As a result, the explicit computation of the mapping induced by $\phi(\cdot)$ could be prohibitively expensive.

The problem can be made tractable by taking a kernel approach that has recently been used to construct nonlinear versions of support vector machines [19], principal components analysis [17], and invariant feature extraction [10,14]. Specifically, the observation behind kernel approaches is that if an algorithm can be written in such a way that only dot products of the transformed data in $\mathcal{F}$ need to be computed, explicit mappings of individual data from $\mathcal{X}$ become unnecessary.

Referring to Equation 9, we know that any column of the solution $\mathbf{V}$, must lie in the span of all training samples in $\mathcal{F}$, i.e., $\mathbf{v}_i \in \mathcal{F}$. Thus, for some $\underline{\alpha} = [\alpha_1, \cdots, \alpha_n]^T$,

$$\mathbf{v} = \sum_{k=1}^{n} \alpha_k \phi(\mathbf{x}_k) = \Phi \underline{\alpha}, \tag{12}$$

where $\Phi = [\phi(\mathbf{x}_1), \cdots, \phi(\mathbf{x}_n)]$. We can therefore project a data point $\mathbf{x}_k$ onto one coordinate of the linear subspace of $\mathcal{F}$ as follows (we will drop the subscript on $\mathbf{v}_i$ in the ensuing):

$$\mathbf{v}^T\phi(\mathbf{x}_k) = \underline{\alpha}^T\Phi^T\phi(\mathbf{x}_k) = \underline{\alpha}^T \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix} = \underline{\alpha}^T\xi_k, \tag{13}$$

where

$$\xi_k = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix}, \tag{14}$$

where we have rewritten dot products, $\langle\phi(\mathbf{x}), \phi(\mathbf{y})\rangle$, with kernel notation, $k(\mathbf{x}, \mathbf{y})$. Similarly, we can project each of the class means onto an axis of the feature space subspace using only dot products:

$$\mathbf{v}^T\mathbf{m}_j = \underline{\alpha}^T\frac{1}{n_j}\sum_{k=1}^{n_j} \begin{bmatrix} \phi^T(\mathbf{x}_1)\phi(\mathbf{x}_k) \\ \vdots \\ \phi^T(\mathbf{x}_n)\phi(\mathbf{x}_k) \end{bmatrix} \tag{15}$$

$$= \underline{\alpha}^T \begin{bmatrix} \frac{1}{n_j}\sum_{k=1}^{n_j} k(\mathbf{x}_1, \mathbf{x}_k) \\ \vdots \\ \frac{1}{n_j}\sum_{k=1}^{n_j} k(\mathbf{x}_n, \mathbf{x}_k) \end{bmatrix} = \underline{\alpha}^T\mu_j. \tag{16}$$

It follows that

$$\mathbf{v}^T S_B \mathbf{v} = \underline{\alpha}^T K_B \underline{\alpha}, \tag{17}$$

where

$$K_B = \sum_{j=1}^{c} n_j(\mu_j - \mu)(\mu_j - \mu)^T, \tag{18}$$

and

$$\mathbf{v}^T S_W \mathbf{v} = \underline{\alpha}^T K_W \underline{\alpha}, \tag{19}$$

where

$$K_W = \sum_{j=1}^{c}\sum_{k=1}^{n_j}(\xi_k - \mu_j)(\xi_k - \mu_j)^T. \tag{20}$$

The goal of Kernel Multiple Discriminant Analysis (KMDA), then, is to find

$$\mathbf{A}_{opt} = \arg\max_{\mathbf{A}} \frac{|\mathbf{A}^T K_B \mathbf{A}|}{|\mathbf{A}^T K_W \mathbf{A}|}, \tag{21}$$

where $\mathbf{A} = [\underline{\alpha}_1, \cdots, \underline{\alpha}_{c-1}]$, and computation of K_B and K_W requires only kernel computations.

4.2 Sampling Data for Efficiency

Because K_B and K_W are $n \times n$ matrices, where n is the size of training set, the nonlinear mapping is dependent on the entire training samples. For large n, the solution to the generalized eigensystem is costly. Approximate solutions could be obtained by sampling representative subsets of the training data, $\{p_k | k = 1, \ldots, M, M < n\}$, and using $\tilde{\xi}_k = [k(\mathbf{x}_1, \mathbf{x}_k), \cdots, k(\mathbf{x}_M, \mathbf{x}_k)]^t$ to take the place of ξ_k.

We select representatives, or *kernel vectors*, by identifying those training samples which are likely to play a key role in $\Xi = [\xi_1, \ldots, \xi_n]$. Ξ is an $n \times n$ matrix, but $rank(\Xi) \ll n$, when the size of training data set is very large. This fact suggests that some training samples could be ignored in calculating kernel features ξ.

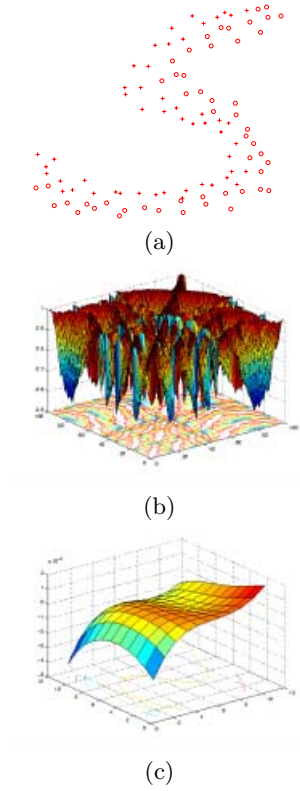


Fig. 1. KMDA with a 2D 2-class non-linearly-separable example. (a) Original data (b) the kernel features of the data (c) the nonlinear mapping.

Our approach is to take advantage of class labels in the data. We maintain a set of kernel vectors at every iteration which are meant to be the key pieces

of data for training. M initial kernel vectors, $KV^{(0)}$, are chosen at random. At iteration k, we have a set of kernel vectors, $KV^{(k)}$, which are used to perform KMDA such that the nonlinear projection $\mathbf{y}_i^{(k)} = \mathbf{V}^{(k)T}\phi(\mathbf{x}_i) = \mathbf{A}_{opt}^{(k)T}\xi_I^{(k)} \in \Delta$ of the original data $\mathbf{x}_i$ can be obtained. We assume Gaussian distribution $\theta^{(k)}$ for each class in the nonlinear discrimination space Δ, and the parameters $\theta^{(k)}$ can be estimated by $\{\mathbf{y}^{(k)}\}$, such that the labeling and training error $e^{(k)}$ can be obtained by $\bar{l}_i^{(k)} = \arg\max_j p(l_j|\mathbf{y}_i, \theta^{(k)})$.

If $e^{(k)} < e^{(k-1)}$, we randomly select M training samples from the correctly classified training samples as kernel vector $KV^{(t+1)}$ at iteration $k+1$. Another possibility is that if any current kernel vector is correctly classified, we randomly select a sample in its topological neighborhood to replace this kernel vector in the next iteration. Otherwise, i.e., $e^{(k)} \geq e^{(k-1)}$, and we terminate. The evolutionary kernel vector selection algorithm is summarized below in Figure 2.

Evolutionary Kernel Vector Selection: Given a set of training data $\mathcal{D} = (X, L) = \{(\mathbf{x}_i, l_i), i = 1, \ldots, N\}$, to identify a set of M kernel vectors $KV = \{\nu_i, i = 1, \ldots, M\}$.

```
// Initialization
k = 0; e = ∞; KV^(0) =random_pick(X);
do{
    // Perfrom KMDA
    A_opt^(k) =KMDA(X, KV^(k));
    // Project 𝒳 to Δ
    Y^(k) =Proj(X, A_opt^(k));

    //Bayesian classifier
    Θ^(k) =Bayes(Y^(k), L);
    // Classification
    L̄^(k) =Labeling(Y^(k), Θ^(k));
    // Calculate error
    e^(k) =Error(L̄^(k), L);

    // Select new kernel vectors
    if(e^(k) < e)
        e = e^(k); KV = KV^(k); k++;
        KV^(k) =random_pick({x_i : l̄_i^(k) /=l_i});
    else
        KV = KV^(k-1); break;
    end
}
return KV;
```

Fig. 2. Evolutionary Kernel Vector Selection

4.3 Kernel D-EM Algorithm

We now apply KMDA to D-EM. *Kernel D-EM (KDEM)* is a generalization of linear D-EM, in which instead of a simple linear transformation of the data, KMDA is used to project the data nonlinearly into a feature space where the data is better separated linearly. The nonlinear mapping, $\phi(\cdot)$, is implicitly determined by the kernel function, which must be determined in advance. The transformation from the original data space $\mathcal{X}$ to the discrimination space Δ, which is a linear subspace of the feature space $\mathcal{F}$, is given by $\mathbf{V}^T\phi(\cdot)$ implicitly or $\mathbf{A}^T\xi$ explicitly. A low-dimensional generative model is used to capture the transformed data in Δ.

Empirical observations suggest that the transformed data often approximates a Gaussian in Δ, and so in our current implementation, we use low-order Gaussian mixtures to model the transformed data in Δ. Kernel D-EM can be initialized by selecting all labeled data as kernel vectors, and training a weak classifier based on only unlabeled samples.

5 Experiments

In this section, we compare KMDA with other supervised learning techniques on some standard data sets. Experimental results of D-EM on content-based image retrieval and view-independent hand posture recognition are presented.

5.1 Benchmark Test for KMDA

We first verify the ability of KMDA with our data-sampling algorithms. Several benchmark data sets[1] are used in our experiments. The benchmark data has 100 different realizations. In [10], results of different approaches on these data sets have been reported. The proposed KMDA algorithms were compared to a single RBF classifier (RBF), a support vector machine (SVM), AdaBoost, and the kernel Fisher discriminant (KFD) [9]. RBF kernels were used in all kernel-based algorithms.

In Table 1, KMDA-pca is KMDA with PCA selection, and KMDA-evol is KMDA with evolutionary selection, where #-KVs is the number of kernel vectors. The benchmark tests show that the proposed approaches achieve comparable results as other state-of-the-art techniques, in spite of the use of a decimated training set.

5.2 Content-Based Image Retrieval

Using a random subset of the database or even the whole database as an unlabeled data set, the D-EM algorithm identifies some "similar" images to the labeled images to enlarge the labeled data set. Therefore, good discriminating

[1] The standard benchmark data sets in our experiments are obtained from `http://www.first.gmd.de/~raetsch`.

Table 1. Benchmark Test: the average test error as well as standard deviation.

Benchmark	Banana	B-Cancer	Heart	Thyroid	F-Sonar
RBF	10.8±0.06	27.6±0.47	17.6±0.33	4.5±0.21	34.4±0.20
AdaBoost	12.3±0.07	30.4±0.47	20.3±0.34	4.4±0.22	35.7±0.18
SVM	11.5±0.07	26.0±0.47	16.0±0.33	4.8±0.22	32.4±0.18
KFD	10.8±0.05	25.8±0.46	16.1±0.34	4.2±0.21	33.2±0.17
KMDA-evol	10.8±0.56	26.3±0.48	16.1±0.33	4.3±0.25	33.3±0.17
#-KVs	120	40	20	20	40

features could be automatically selected through this enlarged training data set to better represent the implicit concepts. The application of D-EM to image retrieval is straightforward. In our current implementation, in the transformed space, both classes are represented by a Gaussian distribution with three parameters, the mean μ_i, the covariance $\mathbf{\Sigma}_i$ and *a priori* probability of each class P_i. The D-EM iteration tries to boost an initial weak classifier.

In order to give some analysis and compare several different methods, we manually label an image database of 134 images, which is a subset of the COREL database. All images in the database have been labeled by their categories. In all the experiments, these labels for unlabeled data are only used to calculate classification error.

To investigate the effect of the unlabeled data used in D-EM, we feed the algorithm a different number of labeled and unlabeled samples. The labeled images are obtained by relevance feedback. When using more than 100 unlabeled samples, the error rates drop to less than 10%. From Figure 3, we find that D-EM brings about 20% to 30% more accuracy. In general, combining some unlabeled data can largely reduce the classification error when labeled data are very few.

Our algorithm is also tested by several large databases. The COREL database contains more than 70,000 images over a wide range of more than 500 categories with 120×80 resolution. The VISTEX database is a collection of 832 texture images. Satisfactory results are obtained.

5.3 View-Independent Hand Posture Recognition

Next, we examine results for KDEM on a hand gesture recognition task. The task is to classify among 14 different hand postures, each of which represents a gesture command mode, such as navigating, pointing, grasping, etc. Our raw data set consists of 14,000 unlabeled hand images together with 560 labeled images (approximately 40 labeled images per hand posture), most from video of subjects making each of the hand postures. These 560 labeled images are used to test the classifiers by calculating the classification errors.

Hands are localized in video sequences by adaptive color segmentation and hand regions are cropped and converted to gray-level images[21]. Gabor wavelet filters with 3 levels and 4 orientations are used to extract 12 texture features.

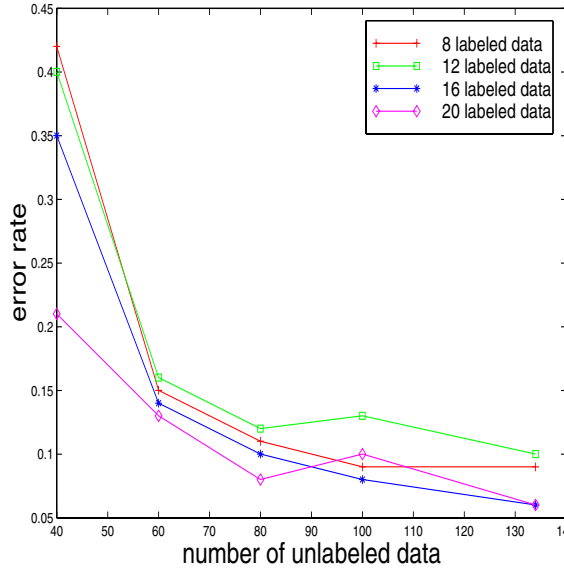


Fig. 3. The effect of labeled and unlabeled data in D-EM. Error rate decreases when adding more unlabeled data. Combining some unlabeled data can largely reduce the classification error.

10 coefficients from the Fourier descriptor of the occluding contour are used to represent hand shape. We also use area, contour length, total edge length, density, and 2nd moments of edge distribution, for a total of 28 low-level image features (I-Feature). For comparison, we also represent images by coefficients of the 22 largest principal components of the total data set resized to 20×20 pixels (these are "eigenimages", or E-Features) [21]. In our experiments, we use 140 (10 for each) and 10000 (randomly selected from the whole database) labeled and unlabeled images respectively, for training with both EM and D-EM. Table 2 shows the comparison.

Table 2. View-independent hand posture recognition: Comparison among multilayer perceptron (MLP),Nearest Neighbor with growing templates (NN-G), EM, linear D-EM (LDEM) and KDEM

Algorithm	MLP	NN-G	EM	LDEM	KDEM
I-Feature	33.3%	15.8%	21.4%	9.2%	5.3%
E-Feature	39.6%	20.3%	20.8%	7.6%	4.9%

We observed that multilayer perceptrons are often trapped in local minima and nearest neighbor suffers from the sparsity of the labeled templates. The poor

performance of pure EM is due to the fact that the generative model does not capture the ground-truth distribution well, since the underlying data distribution is highly complex. It is not surprising that LDEM and KDEM outperform other methods, since the D-step optimizes separability of the classes. Finally, note the effectiveness of KDEM. We find that KDEM often appears to project classes to approximately Gaussian clusters in the transformed space, which facilitates their modeling with Gaussians.

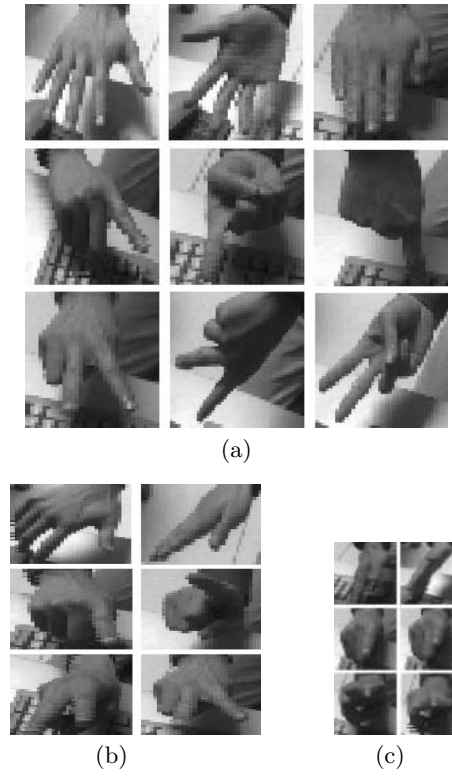


Fig. 4. (a) Some correctly classified images by both LDEM and KDEM (b) images that are mislabeled by LDEM, but correctly labeled by KDEM (c) images that neither LDEM or KDEM can correctly labeled.

6 Conclusion and Future Work

Many visual learning tasks are confronted by some common difficulties, such as the lack of a large number of supervised training data, and learning in high dimensional space. In this paper, we presented a self-supervised learning technique, Discriminant-EM, which employs both labeled and unlabeled data in training,

and explores most discriminant features automatically. Both linear and nonlinear approaches were investigated. We also presented a novel algorithm for efficient kernel-based, nonlinear, multiple discriminant analysis (KMDA). The algorithm identifies "kernel vectors" which are the defining training data for the purposes of classification. Benchmark tests show that KMDA with these adaptations performs comparably with the best known supervised learning algorithms. On real experiments for recognizing hand postures and content-based image retrieval, D-EM outperforms naïve supervised learning and existing semi-supervised algorithms.

Examination of the experimental results reveals that KMDA often maps data sets corresponding to each class into approximately Gaussian clusters in the tranformed space, even when the initial data distribution is highly non-Gaussian. In future work, we will investigate this phenomenon more closely.

Acknowledgments

This work was supported in part by National Science Foundation Grants CDA-96-24396 and IRI-96-34618 and NSF Alliance Program.

References

1. R. Basri, D. Roth, and D. Jacobs. Clustering appearances of 3D objects. In *Proc. of IEEE Conf. Computer Vision and Pattern Recognition*, pages 414–420, 1998.
2. P. Belhumeur, J. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *Proc. of European Conference on Computer Vision*, April 1996.
3. K. Bennett. Combining support vector and mathematical programming methods for classification. In *Advances in Kernel Methods - Support Vector Learning*. Cambridge, MA: MIT Press, 1999.
4. Y. Cui and J. Weng. Hand segmentation using learning-based prediction and verification for hand sign recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 88–93, 1996.
5. R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.
6. A. Gammerman, V. Vapnik, and V. Vowk. Learning by transduction. In *Proc. of Conf. Uncertainty in Artificial Intelligence*, pages 148–156, 1998.
7. T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. of Int'l Conf. on Machine Learning*, 1999.
8. B. Manjunath and W. Ma. Texture features for browsing and retrieval of image data. *IEEE Trans. on Pattern Ananlysis and Machine Intelligence*, 18:837–841, Nov. 1996.
9. Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. Fisher discriminant analysis with kernels. In *IEEE workshop on Neural Networks for Signal Proceesing*, 1999.
10. Sebastian Mika, Gunnar Rätsch, Jason Weston, Bernhard Schölkopf, Alex Smola, and Klaus-Robert Müller. Invariant feature extraction and classification in kernel spaces. In *Advances in Neural Information Processing Systems*, Denver, Nov. 1999.

11. Tom Mitchell. The role of unlabeled data in supervised learning. In *Proc. Sixth Int'l Colloquium on Cognitive Science*, Spain, 1999.
12. K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 1999.
13. M. Popescu and P. Gader. Image content retrieval from image database using feature integration by choquet integral. In *Proc. SPIE Storage and Retrieval for Image and Video Database*, volume VII, 1998.
14. Volker Roth and Volker Steinhage. Nonlinear discriminant analysis using kernel functions. In *Advances in Neural Information Processing Systems*, Denver, Nov. 1999.
15. Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra. Relevance feedback: A power tool for interactive content-based image retrieval. *IEEE Circuits and Systems for Video Technology*, 8:644–655, 1998.
16. S. Santini and R. Jain. Similarity measures. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:871–883, 1999.
17. Bernhard Schölkopf, Alexander Smola, and Klaus Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
18. D. Swets and J. Weng. Hierarchical discriminant analysis for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21:386–400, 1999.
19. V. Vapnik. *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.
20. M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 101–108, Hilton Head Island, South Carolina, 2000.
21. Ying Wu and Thomas S. Huang. View–independent recognition of hand postures. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume II, pages 88–94, Hilton Head Island, South Carolina, June 2000.
22. Ying Wu, Qi Tian, and Thomas S. Huang. Discriminant-EM algorithm with application to image retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, volume I, pages 222–227, Hilton Head Island, South Carolina, June 2000.
23. Ying Wu, Kentaro Toyama, and Thomas S. Huang. Self-supervised learning for object recognition based on kernel Discriminant-EM algorithm. In *Proc. IEEE Int'l Conference on Computer Vision*, Vancouver, 2001.

PCA-Based Model Selection and Fitting for Linear Manifolds

Atsushi Imiya and Hisashi Ootani

Institute of Media and Information Technology, Chiba University
1-33 Yayoi-cho, Inage-ku, 263-8522, Chiba, Japan
{imiya, hisashi}@media.imit.chiba-u.ac.jp

Abstract. We construct an artificial neural network which achieves model selection and fitting concurrently if models are linear manifolds and data points distribute in the union of finite number of linear manifolds. For the achievement of this procedure, we are required to develop a method which determines the dimensions and parameters of each model and estimates the number of models in a data set. Therefore, we separate the method into two steps, in the first step, the dimension and the parameters of a model are determined applying the PCA for local data, and in the second step, the region is expanded using an equivalence relation based on the parameters. Our algorithm is also considered to be a generalization of the Hough transform which detects lines on a plane, since a line is a linear manifold on a plane.

1 Introduction

Independent Component Analyzer (ICA) separates the mean-zero random point distributions in a vector space to a collection of linear subspaces [1]. As an extension of ICA, it could be possible to separate a point set into a collection of linear manifolds whose centroid are uniform, if the centroid of data points are predetermined. In this paper, using the Principal Component Analyzer (PCA) [2,3] we develop an algorithm which separates linear manifolds if the centroids of them are not uniform. We evaluate the performance of this algorithm for the model selection and fitting of a collection of linear manifolds in a vector space.

The PCA is a model of artificial neural networks which solves the eigenvalue problem of a moment matrix of random points in a vector space. In the previous paper [4], we proposed a PCA-based mechanism for the detection of dimensionalities and directions of the object from a series of range images in the three-dimensional vector space. Since the PCA determines the principal minor component of the moment matrix of point distribution, the PCA also solves the least-squares model fitting problem [5,6]. Therefore, a combination of the PCA and the random sampling and voting method achieves the model selection and fitting problems concurrently, same as the Hough transform [7]. This idea is applied to the model fitting problem for the point distribution on a plane. However, for this application we are required to assume the number of parameters of models which is equivalent to the dimension of the model.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 278–292, 2001.

In this paper, we construct an artificial neural network which achieves model selection and fitting concurrently if models are linear manifolds and data points are distributed in the union of a finite number of linear manifolds. For the achievement of this procedure, we are required to develop a method for determineing the dimensions and the parameters of each model and estimating the number of models in a data set. Therefore, we separate the method into two steps. in the first step, the dimension and the parameters of a model are determined applying the PCA for a randomly selected subset of data, and in the second step, the region is expanded using an equivalence relation to the parameters. Our method proposed in this paper is considered to be extension of the previous method, which is for the learning of dimensionalities and directions of an object in 3D space [4], to the higher dimensional case with many objects in the region of interest. Furthermore, our algorithm is also considered to be generalization of the Hough transform which detects lines on a plane, since a line is a linear manifold on a plane.

2 ANN-Based Hough Transform

In pattern recognition, data distributed on manifolds in a higher dimensional vector space $\mathbf{R}^n$ are classified. In this paper, we deal with data distributed on a union of a finite number of linear subspaces and a union of a finite number of linear manifolds.

In an n-dimensional vector space, a k-dimensional linear manifold is expressed as

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \tag{1}$$

where $\boldsymbol{x} \in \mathbf{R}^n$ such that $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)^\top$, $\boldsymbol{A} \in \mathbf{R}^{(n-k)\times n}$, and $\boldsymbol{b} \in \mathbf{R}^k$ for $1 \leq k \leq (n-1)$. This expression [1] is equivalent to

$$(\boldsymbol{I} - \boldsymbol{P})\boldsymbol{x} = \boldsymbol{n},\ \boldsymbol{P}\boldsymbol{n} = \boldsymbol{0}, \tag{2}$$

for an orthogonal projector $\boldsymbol{P}$ and constant vector $\boldsymbol{n}$. which determine the tengent space and the normal direction of this manifold, respectively.

For example, an $(n-1)$-dimensional hyperplane in $\mathbf{R}^n$ is expressed as

$$\boldsymbol{a}^\top \boldsymbol{x} = b,\ \boldsymbol{a} \in \mathbf{R}^n,\ b \in \mathbf{R}. \tag{3}$$

Furthermore, for $n = 2$ and $n = 3$, eq. (3) describes a line on a plane and a plane in a space, respectively. Therefore, if we fix the dimension of models to a constant k, the following algorithm detects all models,

$$\boldsymbol{A}_\alpha \boldsymbol{x} = \boldsymbol{b}_\alpha,\ \alpha = 1, 2, \cdots, m, \tag{4}$$

[1] Equation (2) on a plane is expressed as

$$x\cos\theta + y\sin\theta = r,\ r \geq 0,$$

where vector $(\cos\theta, \sin\theta)^\top$ is the unit normal of this line.

in a space without assuming the number of models m [7], as a generalization of the classical Hough transform for line detection.

The PCA proposed by Oja [2,5,7] estimates the orthogonal projection $\boldsymbol{P}$ to a linear subspace which approximates the distribution of data points in higher dimensional vector spaces. In reference [2], Oja *et al* proposed the following recursive form.

Algorithm 1

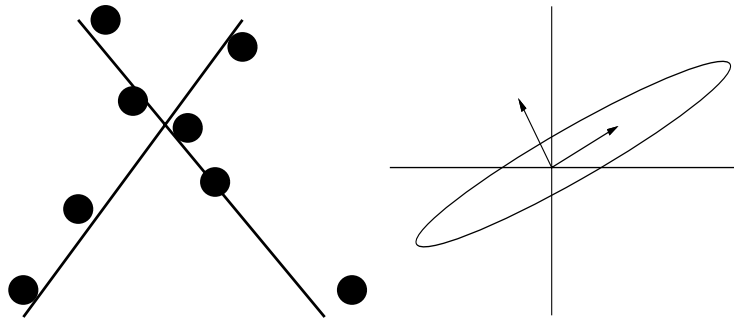
$$\begin{aligned} \tilde{\boldsymbol{W}}(k) &= \boldsymbol{W}(k-1) - \gamma(k)(\boldsymbol{\xi}(k)\boldsymbol{\xi}(k)^{\top})\boldsymbol{W}(k-1) \\ \boldsymbol{W}(k) &= \tilde{\boldsymbol{W}}(k)\boldsymbol{S}(k)^{-1} \\ \boldsymbol{S}(k) &= (\tilde{\boldsymbol{W}}(k)^{\top}\tilde{\boldsymbol{W}}(k))^{1/2}. \end{aligned} \tag{5}$$

This algorithm basically solves the model fitting problems by LSM. Furthermore, the orthogonal projrctor to a space, on which samples lie, is computed as

$$\boldsymbol{P} = \lim_{k\to\infty} \boldsymbol{W}(k)\boldsymbol{W}(k). \tag{6}$$

If we assume that $rank\boldsymbol{P} = 1$, the dimension of the space is one. This property geomtrically means that the PCA detects a line which approximates a distribution of planar points, the mean of which is zero. This algorithm is derived as the steepest decent method which searches the minimum of an energy function. Oja *et al* also extended this idea for the detection of many lines on a plane by combining the algorithm with the self-organization map. This idea is the basis of Algorithm 1 for $n = 2$.

Figure 1 shows the relation between line fitting by the Hough transform and the principal axes of a mean-zero distribution on a plane. The minor component of point distribution determines the direction of the normal vector of a line which passes through the origin.



(a) (b)

Fig. 1. Line fitting and principal cmponent extraction.

Setting $\mathbf{L}$ to be a linear subspace in $\mathbf{R}^n$, a linear manifold is defined as

$$\mathbf{M} = \{\forall \boldsymbol{y} | \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{m},\ \forall \boldsymbol{x} \in \mathbf{L},\ \exists \boldsymbol{m} \in \mathbf{R}^n\}. \tag{7}$$

We say that a linear manifold defined by eq. (7) is parallel to linear subspace $\mathbf{L}$. Setting $\boldsymbol{x}^{\perp}$ to be the vector which is also orthogonal to linear subspace $\mathbf{L}$, vector $\boldsymbol{y}$ on $\mathbf{M}$ is uniquely decomposed as

$$\boldsymbol{y} = \boldsymbol{x} + \boldsymbol{x}^{\perp},\ \boldsymbol{x}^{\top}(\boldsymbol{x}^{\perp}) = 0,\ \boldsymbol{x} \in \mathbf{L},\ \exists \boldsymbol{x} \in \mathbf{R}^n. \tag{8}$$

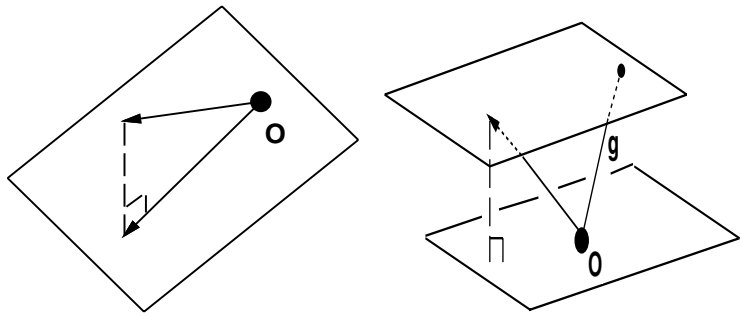
Therefore, setting $\boldsymbol{P}$ to be the orthogonal projector to linear subspace $\mathbf{L}$, a linear manifold parallel to linear subspace $\mathbf{L}$ is described as

$$\mathbf{M} = \{\boldsymbol{y} | \boldsymbol{Q}\boldsymbol{y} = \mu \boldsymbol{n}, \boldsymbol{Q} = \boldsymbol{I} - \boldsymbol{P}\}, \tag{9}$$

where μ is a positive constant and

$$\boldsymbol{n} = \frac{\boldsymbol{Q}\boldsymbol{x}_0}{|\boldsymbol{Q}\boldsymbol{x}_0|} \tag{10}$$

for a vector $\boldsymbol{x}_0$ on linear manifold $\mathbf{M}$. Here, $\boldsymbol{n}$ is a unit vector in linear subspace $\mathbf{L}^{\perp}$, which is the orthogonal compliment of $\mathbf{L}$. Figure 2 shows a linear subspace and a linear manifold which is parallel to a linear subspace.



(a) (b)

Fig. 2. Orthogonal projection to a linear subspace (a) and a linear manifold parallel to a linear subspace (b).

Since the PCA determines the orthogonal projector $\boldsymbol{P}$ and the number of nonzero eigenvalues determines the dimension of a linear subspace, we can use the PCA [2,4] and the recursive form proposed in [4]

$$\boldsymbol{\Lambda}(i+1) = \boldsymbol{\Lambda}(i) + \boldsymbol{D}(i) \tag{11}$$

which computes the eigenvalues of a moment matrix of a mean-zero point distribution. $\boldsymbol{D}(i)$ is the matrix with only diagonal elements of $\boldsymbol{D}$,

$$\boldsymbol{D} = (\boldsymbol{A}(i) + \boldsymbol{A}(i)^\top) + (\boldsymbol{B}(i) + \boldsymbol{B}(i)^\top) + \boldsymbol{C}(i)\boldsymbol{C}(i)^\top, \tag{12}$$

where

$$\begin{aligned} \boldsymbol{A}(i) &= \boldsymbol{\Delta}^\top \boldsymbol{W}(i)\boldsymbol{\Lambda}(i), \\ \boldsymbol{B}(i) &= \boldsymbol{\Delta}^\top \boldsymbol{R}(i)\boldsymbol{W}(i), \\ \boldsymbol{C}(i) &= \boldsymbol{W}(i)^\top \boldsymbol{x}(i+1), \\ \boldsymbol{R}(i) &= \boldsymbol{x}(i)\boldsymbol{x}(i)^\top, \end{aligned} \tag{13}$$

assuming that the mean of points $\{\boldsymbol{x}(i)\}$ is zero, for the sequence of orthogonal matrices computed in Algorithm 2, with

$$\boldsymbol{W}(i+1) = \boldsymbol{W}(i) + \boldsymbol{\Delta}. \tag{14}$$

In these algorithms, grouping of sample points as

$$\mathbf{M} = \mathbf{M}_1 \bigcup \mathbf{M}_2 \bigcup \cdots \bigcup \mathbf{M}_m \tag{15}$$

is achieved by voting. In these algorithms, we assume that the dimensions of space $\mathbf{M}_\alpha$, for $\alpha = 1, 2, \cdots, m$ are all uniform, although we do not predeterminate the number of partitions of a space.

3 Model Selection and Fitting

In this section, we deal with the case in which the dimensions of linear subspaces and linear manifolds are nonuniform and unknown. Therefore our problem is described as follows,

Problem
Assuming that sample points lie in the set

$$\mathbf{M} = \mathbf{M}_1 \bigcup \mathbf{M}_2 \bigcup \cdots \bigcup \mathbf{M}_m$$

for

$$\mathbf{M}_\alpha = \{\boldsymbol{x} | \boldsymbol{A}_\alpha \boldsymbol{x} = \boldsymbol{b}_\alpha, \boldsymbol{A}_\alpha \in \mathbf{R}^{(n-k(\alpha))\times n}, \boldsymbol{b}_\alpha \in \mathbf{R}^{(n-k(\alpha))}\},$$

determine $k(\alpha)$, $\boldsymbol{A}_\alpha$, and $\boldsymbol{b}_\alpha$.

If $\boldsymbol{b}_\alpha = 0$ for $\alpha = 1, 2, \cdots, m$, $\mathbf{M}$ is a union of a finite number of linear subspaces. The first, we derive an algorithm for the detection of linear subspaces if $m = 2$. Then, we extend the method to our main problem described above. For the case of linear subspaces, we assume that the means of random vectors on linear subspaces are zero.

For the problem of a union of a finite number of linear subspaces, the detection of a linear subspace is equivalent to the detection of the orthogonal projector to each space. The orthogonal projector to the space on which samples $\{\boldsymbol{x}\}_{i=1}^{n(\alpha)}$ lie is constructed as

$$\boldsymbol{P}_\alpha = \sum_{i=1}^{k(\alpha)} \boldsymbol{u}_i \boldsymbol{u}_i^\top, \tag{16}$$

where vectors $\{\boldsymbol{u}_i\}_{i=1}^{k(\alpha)}$, are the normalized eigenvectors of correlation matrix

$$\boldsymbol{M}_\alpha = \frac{1}{n(\alpha)} \sum_{i=1}^{n(\alpha)} \boldsymbol{x}_{i\alpha} \boldsymbol{x}_{i\alpha}^\top \tag{17}$$

and $k(\alpha)$ is the rank of matrix $\boldsymbol{M}_\alpha$ such that $k(\alpha) < n$, if the mean of $\{\boldsymbol{x}_{i\alpha}\}_i^{n(\alpha)}$ is zero.

In general, orthogonal projector $\boldsymbol{P}$ computed in eq. (16) does not determine the orthogonal projector to linear subspace $\mathbf{L}_1$ and linear subspace $\mathbf{L}_2$, if two linear subspaces $\mathbf{L}_1$ and $\mathbf{L}_2$ do not contain the others as a subset. Mathematically, for a vector $\boldsymbol{y} \in \mathbf{L}_1$ and a vector $\boldsymbol{z} \in \mathbf{L}_2$, the projector of eq. (16) does not satisfy the relations $\boldsymbol{P}\boldsymbol{y} = \boldsymbol{y}$ and $\boldsymbol{P}\boldsymbol{z} = \boldsymbol{z}$. In the following section, we develop a method to separate $\mathbf{L}_1$ and $\mathbf{L}_2$. Figures 3 (b) and (d) shows the principal axes of a mean-zero point distribution and the principal axes of a union of mean-zero point distributions on a plane and in a space, respectively.

For the separation of two subspaces, we determine the orthogonal projector using a subset of sample points in a finite region. For a finite number of samples $\mathbf{D} = \{\boldsymbol{x}_j\}_{j\in I}$ from $\{\boldsymbol{x}_i\}_{i=1}^n$where I is a subset of $1 \leq i \leq n$, setting $\boldsymbol{g}$ to be the centroid of $\mathbf{D}$, the centroid of vectors

$$\boldsymbol{y}_j = \boldsymbol{x}_j - \boldsymbol{g},\ j \in I \tag{18}$$

is zero. Therefore, applying the PCA to a collection of points $\mathbf{D}_{\boldsymbol{g}} = \{\boldsymbol{y}_j\}_{j\in I}$, we can construct the orthogonal projector $\boldsymbol{P}$ to the linear subspace $\mathbf{L}(\mathbf{D})$ which contains $\mathbf{D}_{\boldsymbol{g}}$. If many samples from $\mathbf{D}$ are contained in linear subspace $\mathbf{L}(\mathbf{D})$, a subset of sample points is distributed on $\mathbf{L}(\mathbf{D})$. Therefore, we first compute the orthogonal projector according to the following algorithm.

Algorithm 2

1 : Select randomly a finite number of samples $\boldsymbol{D} = \{\boldsymbol{x}_j\}_{j\in I}$ from $\{\boldsymbol{x}_i\}_{i=1}^n$ where I is a subset of $1 \leq i \leq n$.

2 : Set $\boldsymbol{y}_j = \boldsymbol{x}_j - \boldsymbol{g}_{\mathbf{D}}$ for vector $\boldsymbol{g}_{\mathbf{D}}$ which is the centroid of $\mathbf{D}$.

3 : Compute the orthogonal projector $\boldsymbol{P}_{\mathbf{D}}$ determined by $\boldsymbol{D}_{\boldsymbol{g}} = \{\boldsymbol{y}_j\}_{j\in I}$.

4 : Accept the linear space $\mathbf{L}(\mathbf{D})$ which corresponds to the projector $\boldsymbol{P}$, If many samples $\boldsymbol{z}_k$ from $\mathbf{D}$ satisfy the relation $\boldsymbol{P}\boldsymbol{z}_k = \boldsymbol{z}_k$.

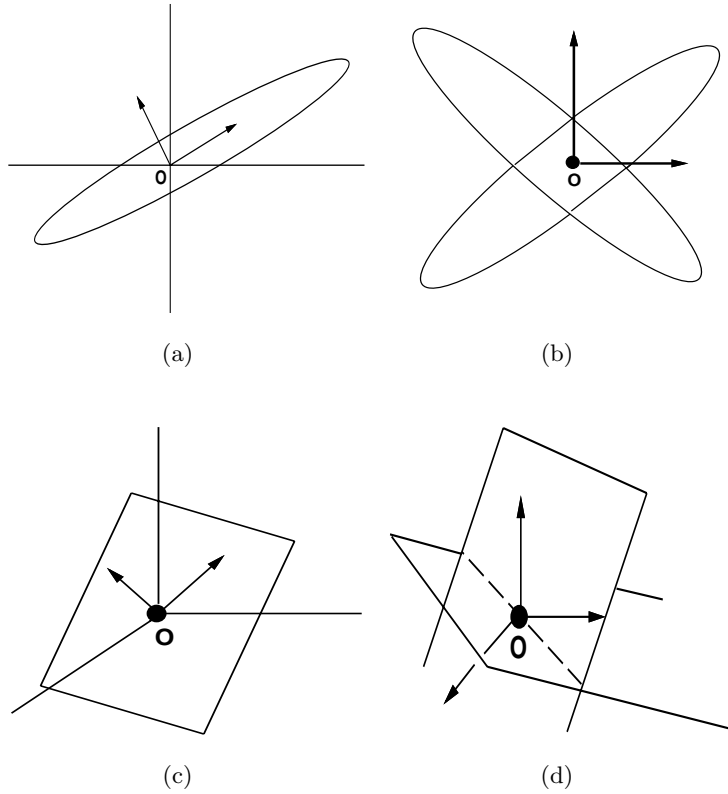


Fig. 3. Point distributions and the directions principal axes: the principal axes of a linear subspace on a plane (a) and in a space (b), and the principal axes of a union of linear subspaces on a plane (c) and in a space (d).

After eliminating points on $\mathbf{L}(\mathbf{D})$, we can detect the other linear subspaces. Figure 4 (a) shows a subset of points on a linear subspace.

Point $\boldsymbol{x}$ on $\mathbf{L}(\mathbf{D})$ holds the relation $\boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}$. This geometric property derives an equivalent relation

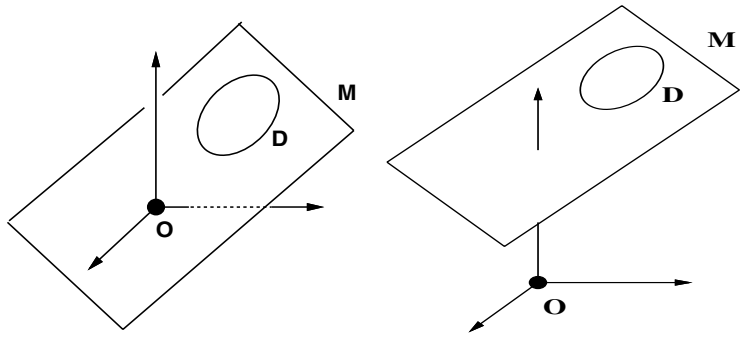
$$\boldsymbol{x} \sim \boldsymbol{y}, \text{ if } \boldsymbol{P}\boldsymbol{y} = \boldsymbol{y} \text{ and } \boldsymbol{P}\boldsymbol{x} = \boldsymbol{x}. \tag{19}$$

We describe the separation of this equivalent relation as

$$[\boldsymbol{x} : \mathbf{L}] = \{\forall \boldsymbol{y} | \boldsymbol{x} \sim \boldsymbol{y}\}. \tag{20}$$

4 Linear Manifold Selection and Fitting

Independent Component Analyzer (ICA) separates the mean-zero random point-distributions in a vector space to a collection of linear subspaces. As an extension



(a) (b)

Fig. 4. A subset on a linear subspace (a) and a subset on a linear manifold parallel to a linear subspace (b).

of ICA, it could be possible to separate a point set into a collection of linear manifolds whose centroid are the same, if the centroid of data points are predetermined. However, if the centroid of each linear manifolds is not same, ICA does not separate manifolds. The algorithm for the separation of linear subspaces does not require the assumption that the mean of sample points of each subspace is zero, since we first compute the projector from a subset of sample points by translating the centroid of a subset to the origin of a vector space. This mathematical property implies that the algorithm for the separation of linear subspaces achieves the separation of linear manifolds, using the equivalent relation

$$\boldsymbol{x} \sim \boldsymbol{y}, \text{ if } \boldsymbol{Q}\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{g}_{\mathbf{D}} \text{ and } \boldsymbol{Q}\boldsymbol{x} = \boldsymbol{Q}\boldsymbol{g}_{\mathbf{D}}. \tag{21}$$

We write a set of points which is equivalent to points on $\mathbf{D}$ as

$$[\boldsymbol{x} : \mathbf{M}] = \{\forall \boldsymbol{y} | \boldsymbol{x} \sim \boldsymbol{y}\}. \tag{22}$$

Therefore, the following algorithm separates sample points to linear manifolds.
Algorithm 3

1 : Set sample points as $\mathbf{S} = \{\boldsymbol{x}_i\}_{i=1}^n$.

2 : Select a finite number of samples $\mathbf{D} = \{\boldsymbol{x}_j\}_{j \in I}$ randomly from $\{\boldsymbol{x}_i\}_{i=1}^n$ where I is a subset of $1 \leq i \leq n$.

3 : Set $\boldsymbol{y}_j = \boldsymbol{x}_j - \boldsymbol{g}_{\mathbf{D}}$ for vector $\boldsymbol{g}_{\mathbf{D}}$ which is the centroid of $\mathbf{D}$.

4 : Compute the orthogonal projector $\boldsymbol{P}_{\mathbf{D}}$ determined by $\boldsymbol{D}_{\boldsymbol{g}} = \{\boldsymbol{y}_j\}_{j \in I}$.

5 : Accept a linear manifold determined as

$$\mathbf{M} = \{\boldsymbol{x} | \boldsymbol{x} = \boldsymbol{P}_{\mathbf{D}}\boldsymbol{y} + \boldsymbol{P}_{\mathbf{D}}\boldsymbol{g}_D\},$$

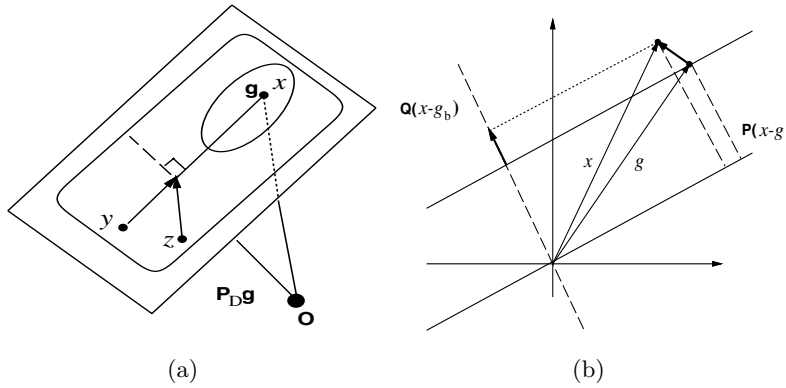
if many samples $\boldsymbol{z}_k$ $k \in J$ form $\mathbf{D}$ hold the relation $\boldsymbol{Q}\boldsymbol{z}_k = \boldsymbol{Q}\boldsymbol{g}_{\mathbf{D}}$.

6 : Expand $\mathbf{D}$ using the equivalent relation $\boldsymbol{x} \sim \boldsymbol{y}$, if $(\boldsymbol{I} - \boldsymbol{P}_{\mathbf{D}})\boldsymbol{x} = (\boldsymbol{I} - \boldsymbol{P}_{\mathbf{D}})\boldsymbol{g}_{\mathbf{D}}$ and $(\boldsymbol{I} - \boldsymbol{P}_{\mathbf{D}})\boldsymbol{y} = (\boldsymbol{I} - \boldsymbol{P}_{\mathbf{D}})\boldsymbol{g}_{\mathbf{D}}$.

7 : Eliminate all points on $\mathbf{M}$ from $\mathbf{S}$, and go to step **1**.

8 : Repeat the procedure until the point set $\mathbf{S}$ becomes the empty set through the elimination of linear manifolds [2].

Figure 4 (a) shows a subset of points on a linear subspace. Furthermore, Figure 5 shows the expansion of a domain using the equivalent relation defined by an orthogonal projector.



(a) (b)

Fig. 5. Domain expansion using an equivalence relation defined by an orthogonal projector (a). The distribution of errors which evaluate the positions of a linear manifold and the centroid of it (b).

For the detection of the dimensionality of linear subspace $\mathbf{L}$ which is parallel to a linear manifold $\mathbf{D}$, we evaluate the distribution of the eigenvalues of the correlation matrix defined sample points in region $\mathbf{D}$. If $E(r)$ such that

$$E(r) = \frac{\sum_{k=1}^{r} \lambda_k}{\sum_{k=1}^{n} \lambda_k} \tag{23}$$

[2] This elimination procedure of a linear manifold from the set of sample points is equivalent to the back-voting of the Hough transform, which eliminates the detected lines from the image plane.

satisfies the relation $E(r) \geq 1 - \sigma$ for a positive small constant σ, we conclude that the dimension of linear subspace which is parallel to a linear manifold is r. The recursive form defined in eq. (11) for vectors $\boldsymbol{y}_i = \boldsymbol{x}_i - \boldsymbol{g}_{\mathbf{D}}$ computes λ_k for $k = 1, 2, \cdots, n$. Although we do not assume to predeteminate the dimensions of linear subspaces which are parallel to linear manifolds, we assume the dimension of a space, in which data sets lie.

5 Numerical Examples and Performance Analysis

In this section, we evaluate the performance of Algorithm 5 proposed in the previous section. The PCA-based algorithm applied to sample points in region $\mathbf{D}$ converges. However, there is no theoretical method for the selection of region $\mathbf{D}$. Once $\mathbf{D}$ is accepted as a subset of a manifold, the expansion of the seed set $\mathbf{D}$ using the equivalence relation with orthogonal projector also converges. Our algorithm contains a step based on a heuristic search for the selection of seed sets. Furthermore, the practical algorithm contains some parameters in the recursive forms. Therefore, we evaluate the performance of our algorithm using computer-generated samples. From a linear subspace $\mathbf{L}_\alpha$, we generated a linear manifold as

$$\mathbf{M}_\alpha = \{\boldsymbol{y} | \boldsymbol{y} = \boldsymbol{x} + \boldsymbol{g}_\alpha,\ \boldsymbol{x} \in \mathbf{L}_\alpha\}. \tag{24}$$

Therefore, the centroid of $\mathbf{M}_\alpha$ is $\boldsymbol{g}_\alpha$ since the mean of $\mathbf{L}_\alpha$ is zero. From data, we computed the centroid of $\mathbf{M}_\alpha$ using the recursive form

$$\boldsymbol{g}(i+1) = \frac{1}{i+1}(i\boldsymbol{g}(i) + \boldsymbol{y}_{i+1}). \tag{25}$$

Setting $\boldsymbol{P}_\alpha$ and $\boldsymbol{h}$ to be the orthogonal projector to linear subspace $\mathbf{L}_\alpha$ computed using the algorithm derived in the previous section and the centroid using eq. (25), respectively, we evaluate the value

$$E = |(\boldsymbol{I} - \boldsymbol{P})(\boldsymbol{h} - \boldsymbol{g}_\alpha)|^2 - |\boldsymbol{P}(\boldsymbol{h} - \boldsymbol{g}_\alpha)|^2. \tag{26}$$

The first term of eq. (26) becomes zero, if both $\boldsymbol{h}$ and $\boldsymbol{g}_\alpha$ lie on a manifold which is parallel to a linear subspace $\mathbf{L}_\alpha$. Furthermore, if $\boldsymbol{h}$ and $\boldsymbol{g}_\alpha$ close each other, the second term of eq. (26) also becomes zero. Moreover, if $|(\boldsymbol{I} - \boldsymbol{P})(\boldsymbol{h} - \boldsymbol{g}_\alpha)| = |\boldsymbol{P}(\boldsymbol{h} - \boldsymbol{g}_\alpha)|$ then the criterion E becomes zero. The condition $|(\boldsymbol{I} - \boldsymbol{P})(\boldsymbol{h} - \boldsymbol{g}_\alpha)| = |\boldsymbol{P}(\boldsymbol{h} - \boldsymbol{g}_\alpha)|$ for small $|(\boldsymbol{I} - \boldsymbol{P})(\boldsymbol{h} - \boldsymbol{g}_\alpha)|$ and $|\boldsymbol{P}(\boldsymbol{h} - \boldsymbol{g}_\alpha)|$ geometrically means that both errors along a manifold and perpendicular to a manifold are the same value. Statistically, this geometric condition means the errors for the estimated centroid and the normal vector of a linear manifold are in the same order. Therefore, this criterion permits us to evaluate the normal vectors of a manifold, which are determined by the principal miner vectors of the linear subspace parallel to this manifold, and the centroid of point distribution on this manifold, simurteneously. In Figure 5 (b), we show the configuration of these vectors which evaluate the fitting of linear manifolds for two dimensional case.

In the first example, 10 lines exist in a three-dimensional vector space. On each line, there exist 500 random points in the region $|x| \leq 100$, $|y| \leq 100$, and $|z| \leq 100$, and the variance of sample points on each line is 0.5.

In the second example, 5 lines and 5 planes exist in a three-dimenional vector space. On each manifold, there exist 500 random points in the region $|x| \leq 100$, $|y| \leq 100$, and $|z| \leq 100$, and the variance of sample points on each line is 0.5. Figure 6, we show the result of extracted manifolds, in this case 5 lines and 5 planes in a three-dimensional space.

In the third example, 10 linear manifolds exist in a ten-dimensional vector space. The dimensions of manifolds are 1 to 9. On each manifolds, there exist 500 random points in the region $|x| \leq 100$, $|y| \leq 100$, and $|z| \leq 100$, and the variance of sample points on each line is 0.5.

For these examples, we generated 10 sets of random samples. We show the values of E for each manifold and the avarage of E for each set of samples. In the second and third examples, our algorithm detects the dimensions of linear manifolds, which is the dimension of linear subspaces parallel to manifolds. The avarages of errors are smaller than 0.5, which is the variance of the sample points. This property confirmes that our algorithm detects models and separates the manifolds on which data lie.

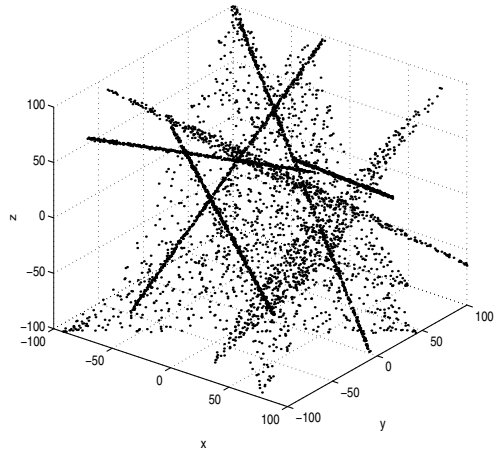


Fig. 6. Extracted lines and planes in 3D space.

6 Conclusions

We have constructed an artificial neural network which achieves model selection and fitting concurrently if models are linear manifolds and data points are distributed in the union of a finite number of linear manifolds. The algorithm is

Table 1. Line fitting in 3D vector space.

model/set	1	2	3	4	5
1	0.026907	0.021519	0.020122	0.757319	0.174356
2	2.518544	0.013811	0.135079	0.223699	0.002123
3	0.170752	0.006271	0.017217	0.166354	0.910127
4	0.450213	0.009680	0.001568	0.064148	0.129328
5	0.050105	0.043180	0.003619	0.040334	1.047528
6	0.083425	0.014767	0.043524	0.047603	0.064470
7	0.550059	0.006188	0.003806	0.352024	0.117675
8	0.011106	0.024073	0.055769	0.091993	0.088041
9	0.246532	0.035074	0.016916	0.142949	0.011740
10	0.621319	0.140549	0.050390	0.067797	0.092995
average	0.472903	0.031511	0.003027	0.200182	0.263839

model/set	6	7	8	9	10
1	0.715922	0.088678	0.022764	0.567278	0.236408
2	0.009577	0.040035	0.101831	0.030510	0.073111
3	1.811411	0.036985	0.011425	0.088330	0.031138
4	0.028323	0.463850	0.183633	0.039801	0.118824
5	0.463369	0.412695	0.172054	0.079120	0.559644
6	0.040622	0.409551	0.021522	0.090134	0.003796
7	0.255582	0.040606	0.126989	0.175285	0.203988
8	0.058565	0.432853	0.201918	0.042971	0.124711
9	0.164041	0.164168	0.493636	0.042458	0.255158
10	0.135181	0.027272	0.048361	0.043834	0.212149
average	0.368260	0.211669	0.138413	0.119972	0.181893

separated in to two steps. The first step of this algorithm determines the dimension and the parameters of a model applying the PCA for local data and the second step of the algorithm expands the region in which sample points hold the equivalence relation to the parameters.

In the previous paper [4], we proposed the PCA-based method for the detection of dimensionalities and directions of the object from a series of range images in the three-dimensional vector space. The method proposed in this paper is considered to be an application of our previous method to the point distribution in the higher dimensional vector space. The performance analysis for a class of computer-generated point distributions confirmed that our method is effective to model separation and fitting in a higher dimensional vector space.

Table 2. Manifold detection in 3D vector space.

model/set	1	2	3	4	5
$*$ 1	0.023276	0.019717	0.008281	0.047246	0.005091
$*$ 2	-0.002459	0.006585	0.073794	0.038131	0.182888
$*$ 3	0.075621	0.022441	0.137228	0.540758	0.006580
$*$ 4	0.005050	0.013835	0.006032	0.072282	0.217258
$*$ 5	0.040343	0.363700	0.104231	0.009294	0.042690
6	0.060595	0.965672	0.031086	0.216097	0.018228
7	0.008472	0.481368	0.083568	0.137222	0.022613
8	0.018566	0.020503	0.002283	0.059936	0.029150
9	0.154401	0.064084	0.135680	0.366759	0.012963
10	0.029011	0.035156	0.025168	0.002276	0.408494
average	0.041288	0.199306	0.060735	0.149000	0.094696

model/set	6	7	8	9	10
$*$ 1	0.365291	0.111762	0.189974	0.140803	0.051011
$*$ 2	0.070667	0.588073	0.321535	0.005803	0.002797
$*$ 3	0.008289	0.049974	0.042725	0.451157	0.888402
$*$ 4	0.128495	0.004209	0.154057	0.079798	0.055014
$*$ 5	0.831821	0.059299	0.007389	0.097214	0.023467
6	0.081076	0.051545	0.050445	0.089089	0.027533
7	0.042650	0.033689	0.009020	0.115958	0.329992
8	0.065655	0.002846	0.080048	0.064180	0.070526
9	0.017523	0.073466	-0.002079	0.179760	0.158162
10	0.010909	0.002074	0.009435	0.008080	0.078100
average	0.162238	0.097694	0.086255	0.123184	0.168500

Symbol $*$ denotes a model is a line.

Let $\mathbf{X}$ be a mean-zero point distribution in $\mathbf{R}^n$. The first principal component $\boldsymbol{u}$ maximizes the criterion

$$J_1 = E_{\boldsymbol{x}\in\mathbf{X}}|\boldsymbol{x}^\top\boldsymbol{u}|^2, \text{ w.r.t, } |\boldsymbol{u}| = 1, \tag{27}$$

where $E_{\boldsymbol{x}\in\mathbf{X}}$ means the expectation over set $\mathbf{X}$. Line $\boldsymbol{x} = t\boldsymbol{u}$ is a one-dimensional linea subspace which approximates $\mathbf{X}$. A maximization criterion

$$J_S = E_{\boldsymbol{x}\in\mathbf{X}}|\boldsymbol{P}_S\boldsymbol{x}|^2, \text{ w.r.t, } \mathrm{rank}\boldsymbol{P}_S = k,\ 2 \leq k < n, \tag{28}$$

determines a k-dimensional linear subspace which approximates $\mathbf{X}$. If the centroid of $\mathbf{X}$ is not predetermined, the maximization criterion

$$J_M = E_{\boldsymbol{x}\in\mathbf{X}}|\boldsymbol{P}_S(\boldsymbol{x}-\boldsymbol{g})|^2, \text{ w.r.t, } \mathrm{rank}\boldsymbol{P}_S = k,\ 2 \leq k < n \tag{29}$$

Table 3. Manifold detection in 10D vector space.

model/set	1	2	3	4	5
1	0.253919(2)	0.215870(2)	0.118065(2)	0.258544(3)	0.591281(1)
2	0.842107(2)	0.290041(3)	0.190744(3)	0.095242(3)	0.386685(2)
3	0.075220(2)	0.090251(3)	0.108855(3)	0.128662(4)	0.240330(3)
4	0.149944(4)	0.264695(3)	0.204970(3)	0.061956(5)	0.151652(5)
5	0.074588(6)	0.114341(3)	0.103091(5)	0.108229(5)	0.060994(5)
6	0.063965(7)	0.097165(3)	0.040622(5)	0.047791(6)	0.056608(6)
7	0.044544(7)	0.142775(4)	0.131182(5)	0.086658(6)	0.047362(7)
8	0.028962(8)	0.075191(4)	0.050028(6)	0.007325(8)	0.012087(8)
9	0.022746(8)	0.063340(6)	0.037107(7)	0.044755(8)	0.033736(8)
10	0.025807(8)	0.023317(7)	0.066194(7)	0.010162(8)	0.001866(9)
average	0.158180	0.137699	0.105086	0.084932	0.158260

model/set	6	7	8	9	10
1	0.165610(1)	0.428742(2)	0.217235(3)	0.211519(1)	0.307913(2)
2	0.524370(1)	0.293748(2)	0.063642(4)	0.564804(1)	0.789513(3)
3	0.268786(3)	0.053212(2)	0.112396(4)	0.297681(2)	0.173967(3)
4	0.069605(4)	0.133380(3)	0.070131(4)	0.099898(2)	0.249656(3)
5	0.116253(5)	0.174491(3)	0.105463(4)	0.689870(2)	0.216040(3)
6	0.073071(5)	0.077947(4)	0.026065(7)	0.291969(2)	0.079472(5)
7	0.063906(6)	0.155310(4)	0.061050(7)	0.049446(6)	0.084121(5)
8	0.025735(8)	0.062035(6)	0.029431(8)	0.046164(7)	0.036738(7)
9	0.011541(9)	0.035130(8)	0.012520(9)	0.030191(8)	0.020935(8)
10	0.009397(9)	0.016899(8)	0.006452(9)	0.014996(9)	0.010756(9)
average	0.132827	0.143089	0.070438	0.229654	0.196911

(♯) expresses the dimension of the linear subspace which is parallel to the linera manifold.

determines a k-dimensional linear manifold which approximates point distribution $\mathbf{X}$. In this paper, we introduced an algorithm for the detection of a collection of linear manifolds.

For an appropriate partition of $\mathbf{X}$ into $\{\mathbf{X}\}_{\mathbf{i}}^{\mathbf{N}}$, such that $\mathbf{X} = \cup_{i=1}^{N} \mathbf{X}_i$, vectors $\boldsymbol{g}_i$ and $\boldsymbol{u}_i$ which maximize the criterion

$$J_C = \sum_{i=1}^{N} E_{\boldsymbol{x} \in \mathbf{X}_i} |(\boldsymbol{x} - \boldsymbol{g}_i)^\top \boldsymbol{u}_i|^2 \tag{30}$$

determines a polygonal curve [8]

$$\boldsymbol{l} = \boldsymbol{g}_i + t\boldsymbol{u}_i, \text{ if } \boldsymbol{l} \in \mathbf{X}_i \tag{31}$$

which approximates $\mathbf{X}$. Furthermore, for an appropriate partition of $\mathbf{X}$ into $\{\mathbf{X}\}_{\mathbf{i}}^{\mathbf{N}}$, such that $\mathbf{X} = \cup_{i=1}^{N}\mathbf{X}_i$, vector $\boldsymbol{g}_i$ and orthogonal projector $\boldsymbol{P}_i$ which maximize the criterion

$$J_P = \sum_{i=1}^{N} E_{\boldsymbol{x}\in\mathbf{X}_i}|\boldsymbol{P}_i(\boldsymbol{x}-\boldsymbol{g}_i)|^2 \tag{32}$$

determines a piecewise flat surface

$$\mathbf{M} = \{\boldsymbol{x}+\boldsymbol{g}_i|\boldsymbol{P}_i\boldsymbol{x}=\boldsymbol{x}\}, \text{ if } \mathbf{M}\subset\mathbf{X}_i \tag{33}$$

which approximates $\mathbf{X}$.

Our criterions for the curve and surface which approximate point distributions are described as the maximization problems for the orthogonal projector and the centroid. Therefore, our method proposed in this paper might be applicable for the detection of curves and surfaces even if many clusters are exist in a data space.

References

1. Karhunen, J., Oja, E., Wnag, L., Vigário, and Joutsensalo, "A class of neural networks for independent component analysis," IEEE Transaction on Neural Networks, **8**, 486-504, 1997.
2. Oja, E., Principal components, minor components, and linear neural networks, *Neural Networks*, **5**, 927-935 1992.
3. Diamantaras, Y. and Kung, S.Y., *Principal Component Neural Networks: Theory and Applications*, John Wiley & Sons, New York, 1996.
4. Imiya, A. and Kawamoto, K., Learning of dimensionality and orientations of 3D objects, Pattern Recognition Letters, **22**, 75-83, 2001.
5. Xu, L., Oja, E. and Suen, C.Y., Modified Hebbian learning for curve and surface fitting, *Neural Networks*, **5**, 441-457, 1992.
6. Heikkonen, J., Recovering 3-D motion parameters from optical flow field using randomized Hough transform, *Pattern Recognition Letters*, **15**, 971-978, 1995
7. Oja, E., Xu, L., and Kultanen, P., Curve detection by an extended self-organization map and related RHT method, Proc. International Neural Network Conference, **1**, 27-30, 1990
8. Hasite, T, and Stuetzle, Prinxipal curves, J. Am. Statistical Assoc, **84**, 502-516, 1989.

Statistics of Flow Vectors and Its Application to the Voting Method for the Detection of Flow Fields

Atsushi Imiya and Keisuke Iwawaki

Institute of Media and Information Technology, Chiba University
1-33 Yayoi-cho, Inage-ku, 263-8522, Chiba, Japan
imiya@media.imit.chiba-u.ac.jp

Abstract. In this paper, we show that the randomized sampling and voting process detects linear flow filed as a model-fitting problem. We introduce a random sampling method for solving the least-square model-fitting-problem using a mathematical property for the construction of pseudo-inverse. If we use an appropriate number of images from a sequence of images, it is possible to detect subpixel motion in this sequence. We use the accumulator space for the unification of these flow vectors which are computed from different time intervals. Numerical examples for the test image sequences show the performance of our method.

1 Introduction

The classical Hough transform estimates the parameters of models. In the classical Hough transformation, the accumulator space is prepared for the accumulation of the voting for the detection of peaks which correspond to the parameters of models to be detected. In this paper, we investigate for the data mining in the accumulator space for the voting method, which is a generalization of the Hough transform, since the peak detection in the Hough transform could be considered as the data discovery in the accumulator space. In this paper, we prepare an accumulator space for for the accumulation of voting of candidate models from many different model spaces. This idea permits us to detect the optical flow field in subpixel accuracy.

In this paper, we deal with the random sampling and voting process for linear flow detection. In a series of papers [1,2], the author introduced the random sampling and voting method for the problems of machine vision. The method is an extension of the randomized Hough transform which was first introduced by Finnish school for planar image analysis [3]. Later they applied the method to planar motion analysis [4] and shape reconstruction from flow field detection [5]. These results indicates that the inference of parameters by voting solves the least-squares problem in machine vision without assuming the predetermination of point correspondences between image frames. We show that the randomized sampling and voting process detects linear flow field. We introduce a new idea to solve the least-square model-fitting problem using a mathematical property for

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 293–306, 2001.

the construction of a pseudoinverse of a matrix. If we use an appropriate number of images from a sequence of images, it is possible to detect subpixel motion in this sequence. In this paper, we use the accumulator space for the unification of flow vectors detected from many time intervals.

The randomized Hough transform is formulated as a parallel distributed model which estimates the parameters of planar lines and spatial planes, which are typical basic problems in computer vision. Furthermore, many problems in computer vision are formulated as model fitting problems in higher dimensional spaces. These problems are expressed in the framework of the least squares method (LSM) for the parameter estimation [6].

Setting $f(x, y, t)$ to be a time-dependent gray-scale image, the linear optical flow $\boldsymbol{u} = (u, v, 1)^\top$ of point $\boldsymbol{x} = (x, y)^\top$ is the solution of the linear equation

$$\boldsymbol{f}^\top \boldsymbol{u} = 0, \tag{1}$$

for

$$\frac{df(x, y, t)}{dt} = \boldsymbol{f}^\top \boldsymbol{u}, \tag{2}$$

where vector $\boldsymbol{f}$ is the spatiotemporal gradient of image $f(x, y, t)$,

$$\boldsymbol{f} = \left(\frac{\partial f(x, y, t)}{\partial x}, \frac{\partial f(x, y, t)}{\partial y}, \frac{\partial f(x, y, t)}{\partial t} \right)^\top. \tag{3}$$

Assuming that the flow vector $\boldsymbol{u}$ is constant in an area Ω, a linear optical flow $\boldsymbol{u} = (u, v, 1)^\top$ is the solution of a system of equations

$$\boldsymbol{f}_\alpha^\top \boldsymbol{u} = 0, \ \alpha = 1, 2, \cdots, N. \tag{4}$$

2 Subpixel Motion

For a sequence of images

$$S_m = \langle f(x, y, -m), f(x, y, -m+1), f(x, y, -m+2), \cdots, f(x, y, 0) \rangle, \tag{5}$$

setting $\boldsymbol{f}_{(k)}$, which is computed from $f(x, y, -k)$ and $f(x, y, 0)$, to be the spatiotemporal gradient between k-frames, we define the k-th flow vector $\boldsymbol{u}_{(k)}$ as the solution of a system of equations

$$\boldsymbol{f}_{(k)\alpha}^\top \boldsymbol{u}_{(k)} = 0, \ \alpha = 1, 2, \cdots, m \tag{6}$$

for each windowed area. From a sequence of images S_m, we can obtain flow vectors $\boldsymbol{u}_{(1)}$, $\boldsymbol{u}_{(2)}$, $\cdots$, $\boldsymbol{u}_{(m)}$, For this example, if we assume the size of a window is $a \times a$, we have ${}_{(a \times a)}C_2 \times m$ constraints among m frames.

Setting $s = kt$, we have the equation,

$$f_x \frac{dx}{ds} + f_y \frac{dy}{ds} + f_s \frac{ds}{dt} = 0. \tag{7}$$

Since $\frac{ds}{dt} = k$, this constraint between the flow vector and the spatiotemporal gradient of an image derives the expression $\boldsymbol{u}_{(k)} = (\frac{dx}{ds}, \frac{dy}{ds}, k)^\top$ for the flow vector detected from a pair of images $f(x, y, (-k + 1))$ and $f(x, y, 0)$. If the speed of an object in a sequence is $1/k$-pixel/frame, the object moves 1 pixel in sequence $S_{(k-1)}$. Therefore, in the spatiotemporal domain, we can estimate the average motion of this point between a pair of frames during the unit time as

$$\overline{\boldsymbol{u}_k} = (\frac{1}{k}u_k, \frac{1}{k}v_k, 1)^\top. \tag{8}$$

form vector $\boldsymbol{u}_{(k)} = (u_k, v_k, k)^\top$.

For the integration of the $\overline{\boldsymbol{u}_k}$ detected the during predetermined time interval, we use the accumulator space. We vote 1 to point $\overline{\boldsymbol{u}_k}$ on the accumulator space for the detection of subpixel flow vectors from a long sequence of images. Therefore, we can estimate the motion of this object from $\{\boldsymbol{u}_{(k)}\}_{k=1}^m$ which is computed from $f(x, y, 1)$ and $f(x, y, m)$. For the unification of vector field $\overline{\boldsymbol{u}_{(k)}}$, we use the accumulator space.

In the accumulator space, we vote $w(k)$ for $\boldsymbol{u}_{(k)}$ for a monotonically decreasing function $w(k)$, such that $w(1) = m$ and $w(m) = 1$. In this paper, we adopt $w(k) = \{(m + 1) - k\}$. This weight of voting means that we define large weight and small weight for short-time motions and long-time motions, respectively.

For the detection of flow vectors at time $t = 0$, traditional methods require the past observations

$$P_m = \{f(x, y, -m), f(x, y, -m + 1), f(x, y, -m + 2), \cdots, f(x, y, 1)\}, \tag{9}$$

the present observation $N = \{f(x, y, 0)\}$, and the future observations,

$$F_m = \{f(x, y, 1), f(x, y, 2), \cdots, f(x, y, m)\}, \tag{10}$$

if methods involves spatiotemporal smoothing. Therefore, the traditional methods involve a process which causes timedelay with respect to the length of the support of a smoothing filter with respect to the time axes.

Our method detects flow vectors of time $t = 0$ using m images $f(x, y, -m + 1)$, $f(x, y, -m + 2)$, $\cdots$, $f(x, y, 0)$, which are obtained for $t \leq 0$, that is, the we are only required data from past. As we will show our method does not require any spatiotemporal preprocessing for this sequence. Our method permits the computetation of flow vectors from past and present data, although the traditional methods with spatiotemporal presmoothing require future data for the computation flow vector. In this sense, our method satisfies the causality of events. Therefore, our method computes flow vectors at time $t = 0$, just after observing image $f(x, y, 0)$. This is one of the advantages of our method. Furthermore, in traditional method, oversmoothing delete slow motions in a sequence of images. Our method preserves slow motions in a sequence of images since the method does not require presmoothing. This is the second advantage of our method.

3 Statistics of Solution of Linear Flow Equation

3.1 Flow Detection by The Hough Transform

The randomized Hough transform is formulated as a parallel distributed model which estimates the parameters of planar lines and spatial planes, which are typical basic problems in computer vision. Furthermore, many problems in computer vision are formulated as model fitting problems in higher dimensional spaces. These problems are expressed in the framework of the least squares method (LSM) for the parameter estimation [6].

Our problem is to estimate a two-dimensionalvector $\boldsymbol{u} = (u, v)^\top$ from a system of equations,

$$a_\alpha u + b_\alpha v = c_\alpha,\ \alpha = 1, 2, \cdots, n. \tag{11}$$

Each equation of this system of equations is considered to be a constraint in a minimization problem of a model-fitting process. Since each constraint determines a line on the u-v plane, the common point of a pair of equations,

$$\boldsymbol{u}_{\alpha\beta} = \{(u, v)^\top | a_\alpha u + b_\alpha v = c_\alpha\} \cap \{(u, v)^\top | a_\beta u + b_\beta v = c_\beta\}, \tag{12}$$

for $\alpha \neq \beta$, is an estimator of the solution which satisfies a collection of constraints. Since we have n constrains, we can have ${}_nC_2$ estimators as the common points of pairs of lines. The estimation of solutions from pairs of equations is mathematically the same procedure as the Hough transform for the detection of lines on a plane from a collection of sample points. Therefore, to speed up the computation time, we can adopt a random sampling process for the selection of pairs of constraints. This procedure derives the same process with the randomized Hough transform.

For the system of equations

$$\boldsymbol{f}_\alpha^\top \boldsymbol{a} = 0,\ \alpha = 1, 2, \cdots, N \tag{13}$$

in a windowed area Ω, we have

$$\boldsymbol{a} = \frac{\boldsymbol{f}_\alpha \times \boldsymbol{f}_\beta}{|\boldsymbol{f}_\alpha \times \boldsymbol{f}_\beta|}. \tag{14}$$

For unit vector $\boldsymbol{a} = (A, B, C)^\top$, the flow vector at the center of the windowed area is computed as

$$\boldsymbol{u} = (\frac{A}{C}, \frac{B}{C}, 1)^\top \tag{15}$$

if C is not zero, since we set $\frac{\delta f(x,y,t)}{\delta t} = 1$. Furthermore, we do not compute the flow vector if C is zero.

3.2 Common Points of a Collection of Lines

For a system of equations

$$\boldsymbol{A}\boldsymbol{x} = \boldsymbol{c}, \tag{16}$$

where

$$\boldsymbol{A} = \begin{pmatrix} a_1, & b_1 \\ a_2, & b_2 \\ \vdots & \vdots \\ a_m, & b_m \end{pmatrix}, \ \boldsymbol{x} = (u, v)^\top, \ \boldsymbol{c} = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_m \end{pmatrix}, \tag{17}$$

and $rank\boldsymbol{A} = 2$, we assume that all row vectors $\{\boldsymbol{a}_i\}_{i=0}^m$ are expressed as

$$\boldsymbol{a}_i = \boldsymbol{a}_1 + \boldsymbol{\delta}_i, i \geq 2, \tag{18}$$

for small vectors $\boldsymbol{\delta}_i$. We call this system of equations an almost singular system. The least-squares solution of the equation

$$\boldsymbol{a}_1^\top \boldsymbol{x} = c_1, \ \boldsymbol{a}_1 = (a_1, b_1)^\top \tag{19}$$

is vector $\boldsymbol{x}_0$ which is perpendicular to line $\boldsymbol{a}_1^\top \boldsymbol{x} = c_1$ and connects the origin and this line. Furthermore, a solution of this system of equations is a common point of lines

$$\boldsymbol{a}_1^\top \boldsymbol{x} = c_1, \ \boldsymbol{\delta}_{ij}^\top \boldsymbol{x} = c_{ij}, \ \boldsymbol{a}_1 = (a_1, b_1)^\top \tag{20}$$

for $\boldsymbol{\delta}_{ij} = (\boldsymbol{\delta}_i - \boldsymbol{\delta}_j)$ and $c_{ij} = (c_i - c_j)$. Therefore, assuming $|\boldsymbol{\delta}_{ij}| \ll 1$ and $|c_{ij}| \ll 1$, the solutions approximately lie on a strip along line $\boldsymbol{a}_1^\top \boldsymbol{x} = c_1$. Therefore, for the accurate estimation of the solution, we adopt the median of points in this strip. This median along a strip is approximated by the average of the medians with respect to arguments u and v. In figure 1, we show a distribution of solutions in a strip along a line on a plane.

Assuming that $\boldsymbol{a}_1^\top \boldsymbol{x} = c_1$ is the linear flow constraint at the center of a windowed area, a collection of linear constraints satisfies the property of an almost singular system of linear equations. Therefore, solutions computed from randomly selected pairs of linear constraints distribute in a strip of finite width along the linear constraint for the centerpoint of the windowed area. Considering this property of the point distribution, we adopt the median of solutions in this strip for each point.

3.3 Motion and Distribution of Solutions

Setting

$$f_{x\alpha} = \left. \frac{\partial f(x, y, t)}{\partial x} \right|_{x=x_\alpha, y=y_\alpha, t=\tau}, \tag{21}$$

$$f_{y\alpha} = \left. \frac{\partial f(x, y, t)}{\partial y} \right|_{x=x_\alpha, y=y_\alpha, t=\tau}, \tag{22}$$

$$f_{\tau\alpha} = \left. \frac{\partial f(x, y, t)}{\partial t} \right|_{x=x_\alpha, y=y_\alpha, t=\tau}, \tag{23}$$

the linear constraint for the linear optical flow for a point $\boldsymbol{x}_\alpha = (x_\alpha, y_\alpha)^\top$ is expressed as

$$f_{x\alpha}u + f_{y\alpha}v + f_{\tau\alpha} = 0. \tag{24}$$

Setting $\boldsymbol{a}$ and $\boldsymbol{a}_i$ to be the spatiotemporal gradient of point $\boldsymbol{x} = (x, y)^\top$ and point $\boldsymbol{x}_i = (x + \alpha_i, y + \beta_i)^\top$, in a windowed area Ω, the flow equation system in this windowed area is given as

$$\boldsymbol{f}_\alpha^\top \boldsymbol{u} = 0,\ \alpha = 1, 2, \cdots, m. \tag{25}$$

Therefore, solutions of the flow equation system in a windowed area Ω distribute on a strip along line

$$\boldsymbol{f}^\top \boldsymbol{u} = 0,\ \boldsymbol{f} = (f_x, f_y, f_t)^\top,\ \boldsymbol{u} = (u, v, 1)^\top, \tag{26}$$

since α_i and β_i are small numbers.

If we define

$$\theta_\alpha = -\frac{f_{x\alpha}}{f_{\tau\alpha}},\ \phi_\alpha = -\frac{f_{y\alpha}}{f_{\tau\alpha}}, \tag{27}$$

for $f_{\tau\alpha} \neq 0$, eq. (24) becomes

$$\frac{u}{\theta_\alpha} + \frac{v}{\phi_\alpha} = 1. \tag{28}$$

This expression of a line for the constraint of flow field implies that vector $(u, v)^\top$ is the common point of lines which connect $(\theta_\alpha, 0)^\top$ and $(0, \phi_\alpha)^\top$, and $(\theta_\beta, 0)^\top$ and $(0, \phi_\beta)^\top$. This property implies that the classical Hough transform achieves the linear-flow field detection voting lines onto the accumulator space. Using this expression, we analyse the performance of the voting method for the detection of flow vectors in windowed areas.

Setting

$$\begin{pmatrix} f'_x \\ f'_y \\ f'_z \end{pmatrix} = \boldsymbol{f} + \boldsymbol{C}\boldsymbol{H}\boldsymbol{e}, \tag{29}$$

where $\boldsymbol{C}$ is a constant matrix, we assume that matric $\boldsymbol{C}$ safisfies the equality $\boldsymbol{C} = \alpha\boldsymbol{I}$ for the identity matrix $\boldsymbol{I}$ and nonzero real constant α, matrix $\boldsymbol{H}$ is the Hessian of spatiotemporal image $f(x, y, t)$, and $\boldsymbol{e} = (1, 1, 1)^\top$. For the equation

$$\frac{u}{-\frac{f'_t}{f'_x}} + \frac{v}{-\frac{f'_t}{f'_y}} = 1, \tag{30}$$

we define parameters A and B as

$$-\frac{f'_t}{f'_x} = -\frac{f_t}{f_x} + A,\ -\frac{f'_t}{f'_y} = -\frac{f_t}{f_y} + B. \tag{31}$$

For parameters A and B, we have the relation

$$A \cdot B = \gamma \begin{vmatrix} \boldsymbol{\tau}^\top \\ \boldsymbol{\alpha}^\top \end{vmatrix} \cdot \begin{vmatrix} \boldsymbol{\tau}^\top \\ \boldsymbol{\beta}^\top \end{vmatrix}, \tag{32}$$

where three vectors are defined as $\boldsymbol{\tau} = (f_t, f_{tt})^\top$, $\boldsymbol{\alpha} = (f_x, f_{xx})^\top$, and $\boldsymbol{\beta} = (f_y, f_{yy})^\top$, and γ is a positive constant. If $A \cdot B < 0$, the common point of the line defined by eq. (30) and line

$$\frac{u}{-\frac{f_t}{f_x}} + \frac{v}{-\frac{f_t}{f_y}} = 1 \tag{33}$$

is close to the vector perpendicular to the line defined by eq. (33). However, if $A \cdot B > 0$, the common point of these two lines is not close to the vector perpendicular to the line defined by eq. (33).

Since we deal with smoothly moving objects, it is possible to assume $|f_{tt}| \ll |f_t|$, that is, we can set $\boldsymbol{\tau} = (f_t, 0)^\top$. This assumption leads to the approximate relation

$$A \cdot B = \gamma f_t^2 \cdot f_{xx} \cdot f_{yy}. \tag{34}$$

Therefore, the sign of $A \cdot B$ is approximately related to the sign of $f_{xx} \cdot f_{yy}$. The second derivatives f_{xx} and f_{yy} describe the smoothness and convexity of the time-varying image $f(x, y, t)$ in the vertical and horizontal directions, respectively. Furthermore, both f_{xx} and f_{yy} approximately describe the local change of the gradient.

If $A \cdot B$ is positive, the gradient does not locally change the direction. If a smooth surface translates, the gradient does not change its direction. Therefore, a typical configuration of vectors $\boldsymbol{\tau}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ for $A \cdot B >$ is yield by a smooth translation. On the other hand, if $A \cdot B$ is negative, the gradient locally changes its direction. If a smooth surface rotates around an axis which is not parallel to the imaging plane, the gradient changes its direction. Therefore, a typical configuration of vectors $\boldsymbol{\tau}$, $\boldsymbol{\alpha}$, and $\boldsymbol{\beta}$ is yield by a rotation. These considerations imply that the LSM method is stable for the detection of rotation. However, LSM is not usually stable for the detection of translation.

4 Detection of Range Flow

If we measure range images, $g(x, y)$, we have the constraint for the spatial motion vector $\boldsymbol{v} = (u, v, w)^\top$, such that,

$$g_x u + g_y v + w + g_t = 0. \tag{35}$$

This system of equations implies that the flow vector lies on a plane in a space. Our problem is to estimate a three-dimensional vector $\boldsymbol{v} = (u, v, w)^\top$ from a system of equations,

$$a_\alpha u + b_\alpha v + c_\alpha w = d_\alpha,\ \alpha = 1, 2, \cdots, n. \tag{36}$$

Since each constraint determines a plane in the u-v-w space, the common point of a triplet of equations,

$$\boldsymbol{v}_{\alpha\beta\gamma} = \bigcup_{i=\alpha,\beta,\gamma} \{(u, v, w)^\top | a_i u + b_i v + c_i w = d_i\} \tag{37}$$

for $\alpha \neq \beta \neq \gamma$, is an estimator of the solution which satisfies a collection of constraints. Since we have n constrains, we can have ${}_nC_3$ estimators as the common points of triples of planes. The estimation of solutions from triplets of equations is mathematically the same procedure as the Hough transform for the detection of planes on in a space from a collection of sample points. Therefore, we can adopt a random sampling process for the selection of triplets of constraints. This procedure derives the same process with the randomized Hough transform for the detection lines on a space. Same as the linear flow filed detection, the solution of range flow distributed along plane. This geometric property of the solutions concludes that the statistical analysis in the accumulator space guarantees the accuracy of the solution.

For a system of equations

$$\boldsymbol{\xi}_\alpha^\top \boldsymbol{a} = 0, \ \alpha = 1, 2, \cdots m, \tag{38}$$

setting

$$\boldsymbol{\Xi} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \cdots, \boldsymbol{\xi}_m)^\top, \tag{39}$$

the rank of matrix $\boldsymbol{\Xi}$ is n if vector $\boldsymbol{x}_\alpha$ is an element of $\mathbf{R}^n$. Therefore, all $n \times n$ square submatrces $\boldsymbol{N}$ of $\boldsymbol{\Xi}$ are nonsingular. Setting N_{ij} to be the ij-th adjacent of matrix $\boldsymbol{N}$, we have the equality

$$f_{\alpha 1} N_{11} + f_{\alpha 2} N_{21} + \cdots + f_{\alpha n} N_{n1} = 0, \tag{40}$$

if the first column of $\boldsymbol{N}$ is $\boldsymbol{\xi} = (\boldsymbol{x}_\alpha^\top, 1)^\top$. Therefore, the solution of this system of equations is

$$\boldsymbol{a} = (n_{11}, n_{21}, \cdots, n_{n1})^\top, n_{i1} = \frac{N_{i1}}{\sqrt{\sum_{j=1}^n N_{j1}^2}}, \tag{41}$$

that is, the solutions distributes on the positive semisphere. For the detection of the range flow filed, we set $n = 3$. Furthermore, for $\boldsymbol{a} = (A, B, C, D)^\top$, we set $\boldsymbol{v} = (\frac{A}{D}, \frac{B}{D}, \frac{C}{D})^\top$ when $D \neq 0$.

If we measure both gray-level and range images, *e.g.*, $f(x, y)$ and $g(x, y)$, we have two constraints for the spatial motion vector $\boldsymbol{u} = (u, v, w)^\top$, such that,

$$g_x u + g_y v + w + g_t = 0, \ f_x u + f_y v + f_t = 0. \tag{42}$$

These system of equations defines a line in the u-v-w space as th common set of points of a pair of planes. Therefore, when we have both images, our search region in the accumulator space is a line in a space. This geometric property of the distribution of solutions reduces the dimension of the accumulator space from two to one. This geometric property speeds up the computation times and reduces the memory-size.

Setting

$$\boldsymbol{g} = \left(\frac{\partial g}{\partial x}, \frac{\partial g}{\partial y}, 1, \frac{\partial g}{\partial t} \right) \tag{43}$$

and $\boldsymbol{e}_3 = (0,0,1)^\top$ we have the relation

$$w = -\boldsymbol{u}^\top \boldsymbol{P}\boldsymbol{g},\ \boldsymbol{P} = \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,0\,1 \end{pmatrix}, \tag{44}$$

for

$$\boldsymbol{u} = \frac{\boldsymbol{f}_\alpha \times \boldsymbol{f}_\beta}{\boldsymbol{e}_3^\top (\boldsymbol{f}_\alpha \times \boldsymbol{f}_\beta)}. \tag{45}$$

Therefore, if we first estimate the field from gray-level image, the depth motion is computed from them. This method also reduces the computational complexity of the range flow filed.

We consider the three-dimensional accumulator space as the discrete space. Let $p(i,j,k)$ be a binary function such that

$$p(i,j,k) = \begin{cases} 1, \text{ if there are votes to point } (i,j,k)^\top, \\ 0, \text{ otherwise.} \end{cases} \tag{46}$$

For

$$u(i) = \sum_{ik} p(i,j,k),\ v(j) = \sum_{jk} p(i,j,k),\ w(k) = \sum_{ij} p(i,j,k), \tag{47}$$

setting $u(a)$, $v(b)$, and $w(c)$ to be the medians of $u(i)$, $v(j)$, and $w(k)$ respectively, we adopt $(a,b,c)^\top$ as the median [1] of $\boldsymbol{v}_{\alpha\beta\gamma}$.

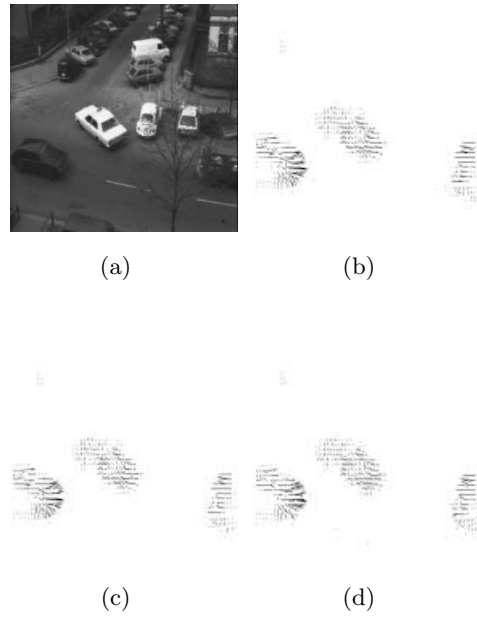
5 Numerical Examples

Setting the size of the windowed area to be 7×7, we have evaluated the effect for the selection of thresholds using frames 1, 2, 3, 4, and 5 of the "**Hamburg Taxi**." Figures 1(a), 1(b), 1(c), and 1(d) show the original image, the flow field detected using all combinations of linear constraints, the flow field detected using randomly selected 30% combinations, and the flow field detected using randomly selected 10% combinations. Here all combinations of linear constraints are ${}_{7\times7}C_2 \times 4$, where 4 is the number of intervals during frames 1, 2, 3, 4, and 5. Figure 1 (d) shows that our method detects all moving objects without artifacts which appear on the background even if we utilized 10% of all linear constrains.

We tested our method for three image sequences, "**Hamburg Taxi**" for frames 0, 1, 2, 3, and 4 in figure 2, "**Rubic Cube**" for frames 7, 8, 9, 10, and 11 in figure 3, and "**SRI tree**" for frames 7,8,9,10, and 11 in figure 4, for the multiframe flow vector detection. In these examples, (a) original image, (b) flow vectors estimated using Lucas and Kanade and, (c) proposed method for 5×5 window using randomly selected 50% constrains.

We compared the results for same images using **Lucas and Kanade** with preprocessing. The preprocessing is summarized as follows [8].

[1] The votes distribute along a plane. We can assume that the number of points whose votes are more than two is small. Therefore, we define a kind of medians in the accumulator space using the median of binary discrete sets [9].



(a) (b)

(c) (d)

Fig. 1. Detected Flow Field. Using frames 1, 2, 3, 4, and 5 of **Hamburg Taxi**. We have evaluated the effect of the selection of thresholds. (a), (b), (c), and (d) are the original image, the flow field detected using all combinations of linear constraints, the flow field detected using randomly selected 30% combinations, and the flow field detected using randomly selected 10% combinations.

- Smoothing using an isotropic spatiotemporal Gaussian filter with a standard deviation of 1.5 pixels-frames.
- Derive the 4-point central difference with mask coefficients $\frac{1}{12}(-1, 8, 0, 8, 1)$.
- The spatial neighborhood is 5×5 pixels.
- The window function is separable in vertical and horizontal directions, and isotropic. The mask coefficients are $(0.00625, 0.25, 0.375, 0.25, 0.00625)$.
- The temporal support is 15 frames.

However, our method does not require any preprocessing. Therefore, we detect the flow field from at least two images.

For "**Hamburg Taxi**," our method detects all motions of a taxi which is turning in the center of this scan and two cars which are crossing the scan in opposite directions. Furthermore, the method detects the subpixel motion of a pedestrian using five frames without presmoothing. However, we could not detect a walking pedestrian using **Lucas and Kanade** method even if we used the 15-frame support.

For "**SRI Tree**," the method detects a tree in a scan as the flow filed which shows the translation of the camera and detects the outline of branches of the

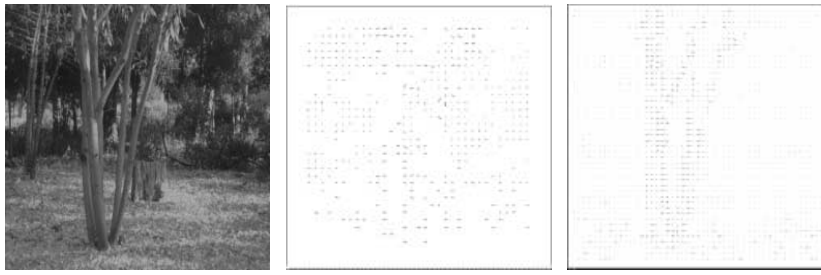
(a) (b) (c)

Fig. 2. Flow Feild of Hamburg Taxi. (a) original image, (b) flow vectors estimated using Lucas and Kanade and, (c) proposed method for 5×5 window using randomly selected 50% constraints.

largest tree. This results means that our method is stable against occlusion if the motion is translation. In this case, the field is the average of the fields detected frame by frame. Furthermore, the method detects motions of both a cube and a turntable in "**Rubic Cube**." These results show that our method is stable to both translation detection and rotation detection in a scan. For the result of "**Rubic Cube**," we can see a uniform background noise. This noise suggests that our method requires improvements of the algorithm for images with large background with the same intensity.

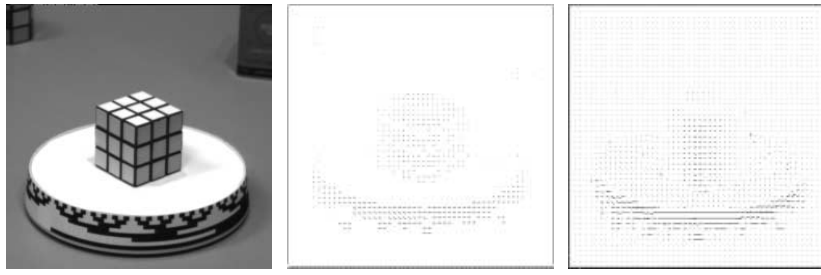
From these numerical results, the performance of our new method without preprocessing is of the same level as **Lucas and Kanade**, which is a very stable method. For the detection of flow vectors, we selected 50% combinations of equations from all possible combinations of a pair of linear equations in the windowed area. The weight for voting is considered as filtering. Therefore, our method involves postprocessing for the detection of motion in a long sequence of images.

In Figure 5, we show the range flow field of a synthetic data. The object is a cone whose bottom is parallel to x-y plane and whose vertex is backward. The cone moves in x-direction. In Figure 5(b) and (c), we illustrated flow vectors in the three-dimensional space and projections of flow vectors to x-y, y-z, and z-y planes, respectively. On the discontinuous edge of the cone, the result shows the errors in the positions of the origins of flow vectors. This problem is caused by the discontinuity of the range data. To improve this problem, we need further analysis.



(a) (b) (c)

Fig. 3. Flow Filed of SRI Tree. (a) original image, (b) flow vectors estimated using Lucas and Kanade and, (c) proposed method for 5×5 window using randomly selected 50% constraints.



(a) (b) (c)

Fig. 4. Flow Field of Rubic Cube. (a) original image, (b) flow vectors estimated using Lucas and Kanade and, (c) proposed method for 5×5 window using randomly selected 50% constraints.

6 Conclusions

We have investigated the possibility of the parameter inference by the integration of data in the accumulator space for the voting method. Our method prepares an accumulator space for the integration of peaks which correspond to the different models.

In this paper, we showed that the random sampling and voting process detects a linear flow field. We introduced a new method of solving the least-squares model-fitting problem using a mathematical property for the construction of a pseudoinverse of a matrix. Furthermore, we showed that using the same mathematical method we can detect the range flow field from a sequence of range images.

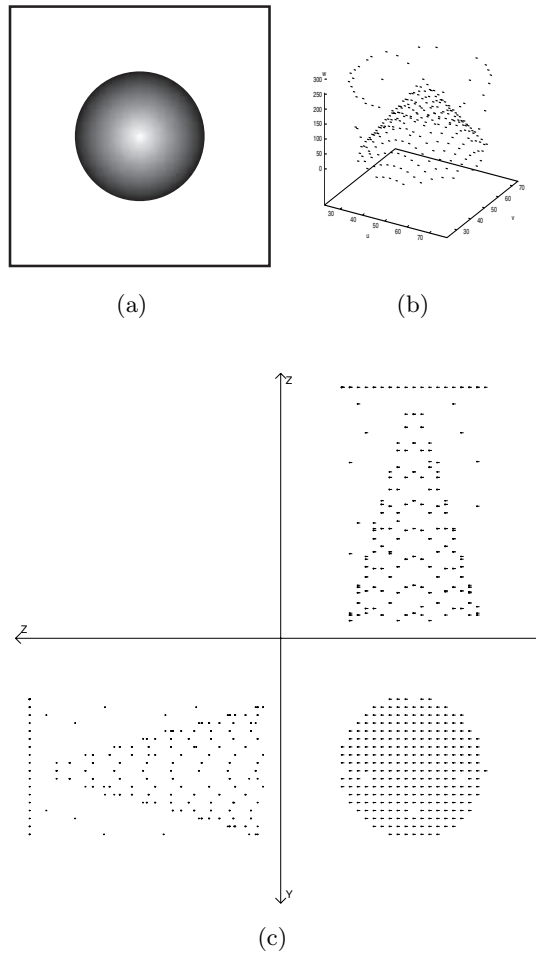


Fig. 5. Range Flow Feild of a Geometric Object. (a) original range image, (b) and (c) flow vectors estimated using random sampling and voting method.

The greatest advantage of the proposed method is simplicity because we can use the same engine for solving multi-constraint problem with the Hough transform for the planar line detection. Our method for the detection of flow vectors is simple because it requires two accumulator spaces for a window, one of which is performed by a dynamic tree, and usually it does not require any preprocessing. Furthermore, the second accumulator space is used for the unification of the flow fields detected from different frame intervals. These properties are advantages for the fast and accurate computation of the flow field.

Acknowledgement

Computer simulation of the range flow detection was performed by Mr. Daisuke Ymada as a part of undergraduate program at the Department of Information and Image Sciences, Chiba University.

References

1. Imiya, A. and Fermin, I. Voting method for planarity and motion detection, *Image and Vision Computing,* **17**, 1999, 867-879.
2. Imiya, A. and Fermin, I., Motion analysis by random sampling and voting process, *Computer Vision and Image Understanding,* **73**, 1999, 309-328.
3. Oja, E., Xu, L., and Kultanen, P., Curve detection by an extended self-organization map and related RHT method, Proc. International Neural Network Conference, **1**, 1990, 27-30.
4. Kälviäinen, H., Oja, E., and Xu, L., Randomized Hough transform applied to translation and rotation motion analysis, *11th IAPR Proceedings of International Conference on Pattern Recognition*, 1992, 672-675.
5. Heikkonen, J., Recovering 3-D motion parameters from optical flow field using randomized Hough transform, *Pattern Recognition Letters*, **15**, 1995, 971-978.
6. Kanatani, K., Statistical optimization and geometric inference in computer vision, *Philosopical Transactions of the Royal Society of London, Series A*, **356** 1997, 1308-1320.
7. Rao, C.R. and Mitra, S.K., *Generalized Inverse of Matrices and its Applications,* John Wiley & Sons, New York. 1971. (Japanese Edition, Tokyo Tosho, Tokyo 1973).
8. Beauchemin, S.S. and Barron, J.L., The computation of optical flow, ACM Computer Surveys **26**, 433-467, 1995.
9. Del Lungo, A., Nivat, M, Pinzani, R., Sorri, L., The medians of discrete sets, Information Processing Letters, **65**, 283-299, 1998

On the Use of Pairwise Comparison of Hypotheses in Evolutionary Learning Applied to Learning from Visual Examples

Krzysztof Krawiec

Institute of Computing Science, Poznań University of Technology,
Piotrowo 3A, 60965 Poznań, Poland
krawiec@cs.put.poznan.pl

Abstract. This paper is devoted to the use of genetic programming for the search of hypothesis space in visual learning tasks. The long-term goal of our research is to synthesize human-competitive procedures for pattern discrimination by means of learning process based directly on the training set of images. In particular, we introduce a novel concept of evolutionary learning employing, instead of scalar evaluation function, pairwise comparison of hypotheses, which allows the solutions to remain incomparable in some cases. That extension increases the diversification of the population and improves the exploration of the hypothesis space search in comparison with 'plain' evolutionary computation using scalar evaluation. This supposition is verified experimentally in this study in an extensive comparative experiment of visual learning concerning the recognition of handwritten characters.

Keywords: visual learning, learning from examples, genetic programming, representation space, outranking relation.

1 Introduction

The processing in contemporary vision systems is usually split into two stages: feature extraction and reasoning (decision making). Feature extraction yields a representation (description) of the original image formulated in terms specified by the system designer. For this purpose, various image processing and analysis methods are being employed [10]. The representation of the analysed image, which is most often a vector of features, is then supplied to the reasoning module. That module applies learning algorithms of statistical or machine learning origin [20,24] to acquire the *knowledge* required to solve the considered task, based on the exemplary data provided by training set of images. That knowledge may have different representation (e.g. probability distributions, decision rules, decision trees, artificial neural networks), depending on the learning algorithm used for its induction.

The most complex part of the design is the search for an appropriate processing and representation of the image data for the considered problem. In most cases the human designer is made responsible for the design, as that issue has been weekly formalized so far (see [10], p.657). This task requires significant extent of knowledge,

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 307–321, 2001.

experience and intuition, and is therefore tedious and expensive. Another difficulty is that the representation chosen by the human expert limits the hypothesis space searched during training of the classifier in the reasoning module and may prevent it from discovering useful solutions for the considered recognition problem.

To overcome these difficulties, in this research we aim at expressing the complete image analysis and interpretation program without splitting it explicitly into stages of feature extraction and classification. The main positive consequence is that the learning process is no more limited to the hypothesis space predefined by the human expert, but encompasses also the image processing and analysis. In other words, we follow here the paradigm of direct approach to pattern recognition, employing the evolutionary computation for the hypothesis space search.

From the machine learning (ML) viewpoint [20,24], we focus on the paradigm of supervised learning from examples as the most popular in the real-world applications. For the sake of simplicity we limit our considerations to the *binary* (two-class) classification problems. However, that does not limit the generality of the method, which may be still applied to multi-class recognition tasks (see Section 5.2.2 for explanation).

This paper is organized as follows. The next section shortly introduces the reader into the metaheuristics of evolutionary programming and, in particular, genetic programming. Section 3 demonstrates some shortcomings of scalar evaluation of individuals in evolutionary computation and proposes an alternative selection scheme based on pairwise comparison of solutions. Section 4 describes the embedding of pairwise comparison into the evolutionary search procedure. Section 5 discusses the use of genetic programming for the search of the hypothesis space in the context of visual learning and presents the results of comparative computational experiment concerning the real-world problem of off-line recognition of handwritten characters. Section 6 discusses the results of the experiments, groups conclusions and outlines the possible future research directions.

2 Evolutionary Computation and Genetic Programming

Evolutionary computation [4,12] has a long tradition of being used for experimental solving of machine learning (ML) tasks [23]. Now it is widely recognized in ML community as a useful search metaheuristics or even as one of ML paradigms [20,24]. It is highly appreciated due to its ability to perform global parallel search of the solution space with low probability of getting stuck in local minima. Its most renowned applications in inductive learning include feature selection [33], feature construction [1], and concept induction [5,7]. In this paper, we focus on the last of the aforementioned problems, with solutions (individuals) implementing particular hypotheses considered by the system being trained; from now on, the terms ‘solution’, ‘individual’ and ‘hypothesis’ will be used interchangeably.

Evolutionary computation conducts an optimization search inspired by the inheritance mechanisms observed in nature. This metaheuristics maintains a set of solutions (individuals), called *population* in evolutionary terms. In each step (generation) of the algorithm, the *fitness* (performance) of all the solutions from the

population is measured by means of the problem-specific scalar evaluation function *f*. Then, some solutions are *selected* from the population to form the 'mating pool' of parents for the next generation. This selection depends on the values of *f* and may follow different schemes, for instance the roulette-wheel principle or tournament selection, to mention the most popular ones (see [7,23] for details). Selected solutions undergo then the *recombination*, which usually consists in exchanging randomly selected parts of the parent solutions (so-called *crossover*). In that process, the useful (i.e. providing good evaluation by *f*) features of the parent solutions should be preserved. Then, randomly chosen offspring solutions are subject to *mutation*, which introduces minor random changes in the individuals. The sequence of evaluation, selection and recombination repeats until an individual having satisfactory value of *f* is found or the number of generations reaches a predefined limit.

Genetic programming (GP) proposed by Koza [15] is a specific paradigm of evolutionary computation using sophisticated solution representation, usually LISP expressions. Such representation is more direct than in case of genetic algorithms, which require the solutions to be represented as fixed length strings over binary alphabet. That feature simplifies the application of GP to real-world tasks, requires however more sophisticated recombination operators (crossover and mutation). GP is reported to be very effective in solving a broad scope of learning and optimization problems, including the impressive achievement of evolving human-competitive solutions for the controller design problems, some of which have been even patented [16].

3 Scalar Evaluation vs. Pairwise Comparison of Hypotheses

3.1 Complete vs. Partial Order of Solutions

Similarly to other metaheuristics, like local search, tabu search, or simulated annealing, the genetic search for solutions requires an existence of an evaluation function *f*. That function guides the search process and is of crucial importance for its final outcome. In inductive learning, *f* should estimate the predictive ability of the particular hypothesis. In the simplest case, it could be just the accuracy of classification provided by the hypothesis on the training set. However, in practice more sophisticated forms of *f* are usually applied to prevent the undesired overfitting phenomenon, which co-occurs with characteristic for GP overgrowth of solutions. One possible remedy is to apply here the multiple train-and-test approach (so-called *wrapper*) on the training set or to introduce an extra factor penalizing too complex hypotheses, either explicitly or in a more concealed manner (as, for instance, in the minimum description length principle [26]).

The primary claim of this paper is that scalar evaluation reflects well the hypothesis utility, reveals however some shortcomings when used for hypothesis comparison. In particular, scalar evaluation imposes a *complete order* on the solution space and therefore forces the hypotheses to be always *comparable*. That seemingly obvious feature may significantly deteriorate the performance of the search, as illustrated in the following example.

Example 1. For a hypothesis h considered by an inductive algorithm, let $C(h)$ denote the subset of examples from a non-empty training set T that it correctly classifies ($C(h) \subseteq T$). Let the hypotheses be evaluated by means of the scalar evaluation function f being the accuracy of classification of h on T, i.e. $f(h) = |C(h)| / |T|$. Let us consider three hypotheses, a, b, and c, for which $|C(a)| > |C(b)| = |C(c)|$. Thus, with respect to f, hypotheses b and c are of the same quality and are both worse than a.

This evaluation method cannot differentiate the comparison of hypotheses (a,b) and (a,c). Due to its aggregating nature, scalar evaluation ignores the more sophisticated mutual relations between a, b and c, for instance the set-theoretical relations between $C(a)$, $C(b)$ and $C(c)$. If, for instance, $C(b) \subset C(a)$, we probably would not doubt the superiority of a over b. But what about the relation between a and c, assuming that $C(c) \not\subset C(a)$ and $|C(c) \cap C(a)| << |C(a)|$? In such a case, although a classifies correctly more examples than c, there is a (potentially large) subset of examples $C(c) \setminus C(a)$, which it does not cope with, while they are successfully classified by c. Thus, superiority of a over c is rather questionable. Moreover, if also $C(a) \not\subset C(c)$, the question concerning mutual relation between a and c should intuitively remain without answer. ■

This example shows us that scalar evaluation applied to hypothesis comparison can show prejudice against hypotheses that are only slightly worse, but significantly different with respect to the 'behavior' on the (training) data. The primary reason for this shortcoming is the aggregating and compensatory nature of the summation operator used, for instance, in the definition of accuracy of classification. Such measures may yield similar or even equal values for very different hypotheses. An important implication for the (e.g. evolutionary) search of hypothesis space is that some novel and 'interesting' hypotheses, which could initiate useful search directions, may be discarded in the search.

This limitation of scalar aggregating measures is well known in multiple-criteria decision aid, where models alternative to the functional one have been elaborated to overcome that difficulty (see, for instance, [32]). Following those ideas, we propose the *relational* method of hypothesis evaluation and selection instead of the *functional* one. In particular, we suggest that when the considered hypotheses 'behave' in a significantly different way on the training set, we should allow them to remain *incomparable*. Allowing incomparability of solutions implies modifying the hypothesis space structure from the *complete* order to the *partial* order. To model such a structure, we propose to use a binary *outranking relation*[1], denoted thereafter by '$\succeq$' (see, for instance, chapter 5 of [32]). For a pair (a,b) of solutions, $a \succeq b$ should express the fact that a is *at least as good* as b. Then, exactly one of the four following cases is possible:

- a is indiscernible with b ($a \succeq b$ and $b \succeq a$), or
- a is strictly better than b ($a \succeq b$ and not $b \succeq a$), or
- b is strictly better than a ($b \succeq a$ and not $a \succeq b$), or
- a and b are incomparable (neither $a \succeq b$ nor $b \succeq a$).

[1] Formally, an outranking relation induces partial *pre*order, as it permits indiscernibility.

Partial order has a natural graphical representation of directed graph. The nodes of an outranking graph correspond to hypotheses, whereas arcs express the outranking. Particularly, the potentially best solutions should not be outranked, and are therefore represented in such graph by initial (predecessor-free) nodes. Note also that outranking is in general reflexive and non-symmetric.

3.2 Hypothesis Outranking Relation for Learning from Examples

At this point we face the need for the choice of a particular form of hypothesis outranking. Because in this study we focus on the paradigm of learning from examples, it seems reasonable to benefit from the existence of the training set.

The idea is to get rid of the aggregating nature of scalar evaluation measures and to go more into detail by *analyzing the behavior of hypotheses on particular instances from the training set*. Intuitively, the need of incomparability grows with the dissimilarity between the compared hypotheses and becomes especially important when their scalar evaluations are relatively close. Example 1 showed us that it makes sense to base the comparison of a pair of hypotheses (a,b) on the set difference of the sets of properly classified instances ($C(a)$ and $C(b)$, respectively). In particular, the more examples belong to $C(b) \setminus C(a)$, the less likely should be the outranking $a \geq b$.

An outranking relation with such properties may be reasonably defined in several different ways. In our previous study on this topic [19], for the sake of simplicity we applied the crisp set inclusion and defined the outranking of a over b as follows:

$$a \geq b \Leftrightarrow C(b) \subseteq C(a) . \quad \textbf{(1)}$$

This definition states that hypothesis a outranks hypothesis b iff a classifiers correctly at least all the examples that are classified correctly by b. Although previous computational experiment showed the usefulness of this simple definition [19], it has a serious drawback of being very sensitive. The outranking of a over b may be disabled by just a single training example, i.e. when $C(b) \setminus C(a) \neq \emptyset$. Thus, in this work we try to relax the crisp condition used in (1). For this purpose we 'fuzzify' in a sense the condition on the right side of definition (1) and refer to the notion of *set inclusion grade* (for overview, see [6]). For a pair of sets A and B, the inclusion grade $I(A,B)$ measures the extent of inclusion of A in B. In particular, we rely here on the inclusion grade as defined by Sanchez [27]:

$$I(A,B) = \frac{|A \cap B|}{|A|} . \quad \textbf{(2)}$$

For any nonempty set A and any set B, $I(A,B) \in \langle 0,1 \rangle$, $I(\emptyset,A)=0$ and $I(A,A)=1$.

Based on this notion we can define now the hypothesis outranking referring to the sets $C(a)$ and $C(b)$ of correctly classified examples:

$$a \geq b \Leftrightarrow I(C(b), C(a)) \geq \eta , \quad \textbf{(3)}$$

where η is a user-defined threshold having an interpretation of the acceptable percentage of $C(b)$ and $C(a)$ intersection, measured in relation to $C(b)$. We expect this definition to be less

sensitive than (1). However, for this advance we pay the price of introduction of an extra parameter η.

The definitions (1) and (3) are not much useful in practice as they do not take into account the class distributions in $C(a)$ and $C(b)$. For instance, the so-called default hypotheses (i.e. hypothesis pointing to the decision class with largest a priori probability for all examples from T) will be quite well protected from being outranked by the non-default ones. This is contradictory to common sense expectation, as the default hypotheses are trivial and the least desired in the search. To overcome that difficulty, we have to make the outranking definition (3) more specific and consider separately the positive and negative decision class (limiting our considerations, without loss of generality, to the binary classification problems). Let T^+ and T^- denote respectively the subsets of positive and negative examples in the training set T ($T^+ \cup T^- = T$, $T^+ \cap T^- = \varnothing$). Then, let $C^+(h) = C(h) \cap T^+$ and $C^-(h) = C(h) \cap T^-$. Finally, we define the outranking in the following way:

$$a \geq b \Leftrightarrow I(C^+(b), C^+(a)) \geq \eta \wedge I(C^-(b), C^-(a)) \geq \eta \,. \quad (4)$$

This definition requires the outranking hypothesis a to be superior to the outranked hypothesis b with respect to both positive and negative classes.

To avoid confusions, it is important to stress that the partial order imposed by hypothesis outranking as defined in (1), (3) or (4) refers to the 'behavior' of the hypothesis on the training data and therefore it has nothing to do with orders based on the hypothesis *representation*, which are also often considered in the literature (e.g. the partial order of decision trees used by top-down decision tree inducers). The approach presented in this paper is universal in the sense that it does not make any assumption about knowledge representation used by the particular induction algorithm.

4 Extending the Evolutionary Learning Procedure by Hypothesis Outranking

4.1 Genetic Programming Using Partial Order of Solutions

This section describes shortly the modifications that need to be introduced into the evolutionary search procedure due to use of pairwise hypothesis comparison. In particular, changes have to be made at all those stages, which make use of the scalar evaluation function, i.e. primarily to the selection process[2]. The evolutionary learning procedure extended in the way described below will be further referred to as *GPPO* (*Genetic Programming using Partial Order of solutions*).

Selection is the central step of any evolutionary programming procedure and consists in picking out the subset P^* of parent solutions from the population P evolved in particular generation of evolutionary search (see Section 2). It is the main factor that implements the so-called evolutionary pressure and influences search

[2] Formally, also the maintenance of the best solutions found in the search should undergo some modifications when the pairwise comparison is used instead of scalar evaluation (see detailed discussion in [19]). We focus here on the selection as the stage of crucial importance for the search convergence and effectiveness.

convergence. In particular, when using the relational model instead of the functional one, we should be ready to handle hypotheses that are incomparable.

The proposed outranking-based selection procedure proceeds as follows. We start with computing the subset $N(P)$ of *non-outranked* solutions from P, i.e.

$$N(P) = \{h \in P: \neg\exists\, h' \in P: h' \geq h\}. \quad \textbf{(5)}$$

This definition is straightforward, but its result is troublesome as we cannot directly control the cardinality of $N(P)$. In practice $N(P)$ usually contains a small fraction of the original population, nevertheless in extreme cases it can be empty or encompass all the individuals from P. This is contradictory to the reasonable assumption that we should preserve constant size of the population (at least approximately).

Thus, the method described in this paper combines the selection of non-outranked solutions with the widely used in evolutionary computation tournament selection [8] in the following steps:

1. Set $P^* \leftarrow N(P)$.
2. If $|P^*|$ is smaller than a predefined fraction $\alpha \in \langle 0,1\rangle$ of the population size $|P|$ ($|P^*| < \alpha|P|$), the solutions in P^* are 'cloned' to reach that size.
3. The missing part of the mating pool ($P \setminus P^*$) is filled with solutions obtained by means of the standard tournament selection [8] on P.

This procedure strengthens the proliferation of non-outranked solutions from P when there are few of them. On the other hand, we avoid the premature convergence of the search exclusively on the non-outranked solutions.

4.2 Related Research

Methods of improving the exploration of the solution space (or maintenance of diversity) appear in evolutionary computation under the name of *niching* and *multimodal* genetic search. Some of those methods operate on the solution level and base the selection on a random, usually small sample of the population (e.g. tournament selection by Goldberg, Deb, and Korb [8], or restricted tournament selection by Harik [11]). Others use a more careful pairing of selected parents ([24] p. 259). Yet another approaches rely on a more intermediate influence and modify the evaluation scheme, penalizing the solutions for 'crowding' in the same parts of the solution space, as in the popular fitness sharing by Goldberg and Richardson [9] or sequential niche technique by Beasley, Bull and Martin [2]. In particular, niches may be maintained during the entire evolution process (parallelly) or only temporarily (sequentially); Mahfoud [22] provided an interesting comparison of these groups of methods.

The specificity of GPPO method in comparison to the aforementioned approaches consists in the following features:

– GPPO supports niching in an explicit way, by means of the concept of outranking. In particular, GPPO does not require any extra distance metric in the search space (whereas, for instance, many fitness sharing methods do).

- GPPO carries out the search without making any reference to the scalar evaluation function, which, as shown in Section 3.1, has some drawbacks due to its aggregative character in machine learning tasks. Thus, GPPO is more than a mere niching method; it is rather a variety of evolutionary search procedure that maintains the set of mutually non-outranking solutions during the search process.
- GPPO makes direct use of the detailed and very basic information on performance of the solution on particular training examples. Thus, the comparisons of individuals in the genetic GPPO search are tied very closely to the mutual relationships of hypotheses in the hypothesis space.

A reader familiar with the topic may notice that some of the ideas presented in this paper are analogous to those of multiobjective genetic search and optimization [28,31]. However, those approaches refer to the dominance relation, which assumes an existence of the multidimensional space spanned over a finite number of ordered objectives. The concept of hypothesis outranking presented in Section 3.2 and, in particular, the outranking definition (4) used in the following case study, do not assume the existence of such a space. The incomparability of solutions in dominance-based methodology is a consequence of the presence of multiple dimensions (objectives) and the tradeoffs between them, whereas it is in general not the case in the outranking relation.

5 Inducing Pattern Recognition Procedures from Examples by Means of Evolutionary Computation

5.1 Representation of Image Analysis Programs

A remarkable part of evolutionary computation and, in particular, genetic programming research concerns machine learning (see [23,24] for review). There are also several reports on applications of genetic metaheuristics in image processing and analysis (e.g. [1]). However, there are relatively few, which try to combine both these research directions and refer to the visual learning, understood as the search for complete pattern analysis and/or recognition programs [14,29,25,17,19].

As stated in Introduction, in this study we aim at expressing the complete image analysis and interpretation program without splitting it explicitly into stages of feature extraction and interpretation. The search takes place in the space of hypotheses being pattern recognition procedures expressed in GPVIS [18]. GPVIS is an image analysis-oriented language encompassing a set of operators responsible for simple feature extraction, region-of-interest selection, as well as arithmetic and logical operations. The programs performing image analysis and recognition are GPVIS expressions composed of such operations. GPVIS allows formulating the complete pattern recognition program without the need for an external machine learning classifier, what is required if the processing is split into the feature extraction module and the reasoning module.

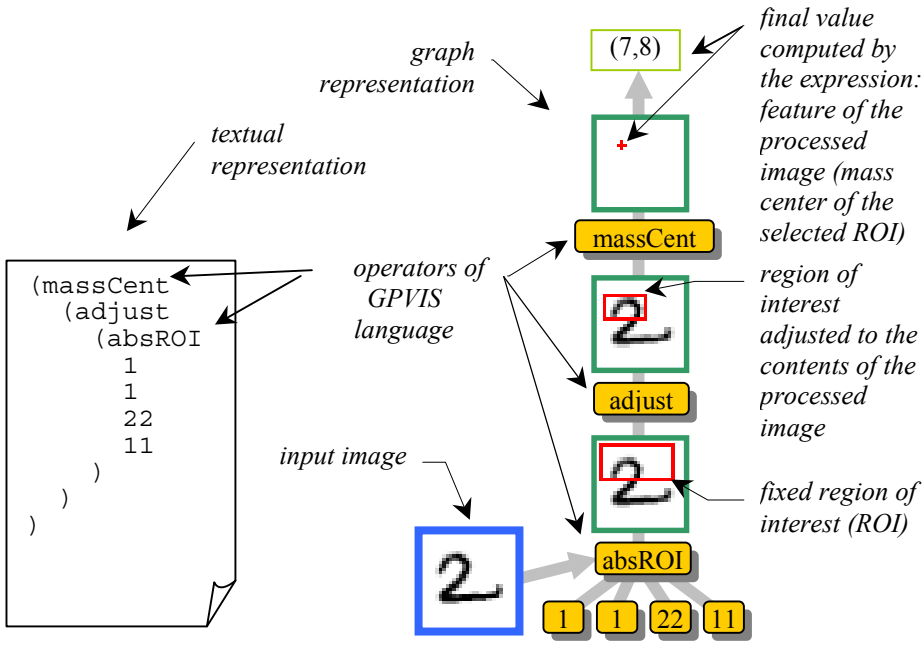


Fig. 1. Tree-like and textual representations of an exemplary GPVIS expression with an illustration of processing on an image of a digit

Figure 1 shows an exemplary GPVIS expression in both textual and graphical form. The picture illustrates also the processing carried out by this expression when applied to an exemplary image of digit '*2*'. In particular, the expression constructs a rectangular region of interest (*absROI 1 1 22 11*), which is then adjusted by GPVIS operator *adjust* to the minimal bounding rectangle based on the image contents. Finally, the *massCent* operator computes the center of the mass of the selected image fragment and that point (a pair of coordinates) constitute the outcome of the computation. This returned value could be then further processed to yield binary class assignment, as it is in the case study described in this section. Detailed description of the GPVIS language may be found in [18].

5.2 The Computational Experiment

The primary goal of the following computational experiment was to compare the search effectiveness of the 'plain' genetic programming (GP) and genetic programming using pairwise comparison of solutions (GPPO) described in Section 3. The main subject of comparison was the accuracy of classification of the best evolved solutions (hypotheses) on the training and test set.

5.2.1 Off-Line Handwritten Character Recognition

As the experimental test bed for the approach, we chose the problem of off-line handwritten character recognition. This task is often referred to in the literature due to

the wide scope of its real-world applications. Proposed approaches involve statistics, structural and syntactic methodology, sophisticated neural networks, or ad hoc feature extraction procedures, to mention only the most known (for review, see [21]).

The source of images was the MNIST database of handwritten digits provided by LeCun et al. [21]. MNIST consists of two subsets, training and test, containing together 70,000 images of digits written by approx. 250 persons (students and clerks), each represented by a 28×28 halftone image (Fig. 2). Characters are centered and scaled with respect to their horizontal and vertical dimensions, however, not 'de-skewed'.

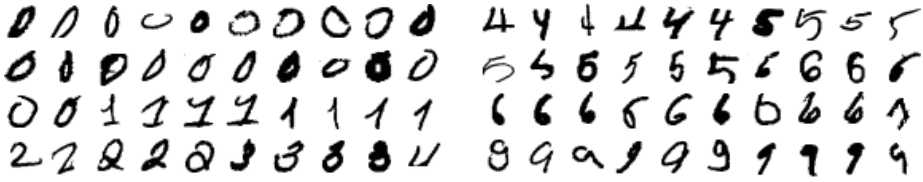


Fig. 2. Exemplary difficult examples selected from the training part of the MNIST database

5.2.2 The Need for a Meta-classifer

According to GPVIS syntax, programs formulated in that language return logical value (true or false), so it is impossible to build the complete 10-class digit recognition system using GPVIS in a direct way. Therefore, we had to decompose the problem into binary classification tasks, where the decision can be computed by an expression written in GPVIS.

Such decomposition may be done in several ways. In this particular experiment we follow the approach that is the most computationally expensive but, as reported in the literature, provides good results [13]. The original ten-class classification problem is decomposed into 10×9/2=45 binary problems, each for one pair of decision classes. The training (here: the evolution) is carried out for each binary subproblem separately, based on the training set limited to the examples representing the two appropriate decision classes. The triangular matrix of 45 independently induced classifiers form the so-called *metaclassifier*, in particular the n^2 type of it [13]. The classification (recognition) of a new object (image) requires querying all the binary classifiers. The final assignment of an image to one of the 10 decision classes is obtained by an appropriate aggregation of decisions made by particular binary classifiers. For details related to the *meta-classifiers* issue the reader should refer to the literature [3].

5.2.3 Experiment Design

In the process of software implementation and experiment preparation we took special care of ensuring comparability of results. The GP and GPPO runs for particular binary problems were 'paired' in the sense that they started from the same initial population and used the same training and test sets as well as the values of parameters: population size: 50; probability of mutation: .05; tournament selection scheme [8] with tournament size equal to 5. In each generation, half of the population was retained unchanged, whereas the other fifty percent underwent recombination.

The training set contained 100 examples (images), 50 for each of two considered decision (digit) classes, selected randomly from the training part of the MNIST database. There was one-to-one correspondence between the original images and examples in the ML sense. The GP runs used the standard tournament selection based on scalar fitness function, whereas GPPO runs followed the selection procedure described in Section 4. The η parameter (see formula (4)) was set to .95, and the proliferation coefficient α (see Section 4.1) to .1 on the ground of preliminary series of experiments.

In the recombination process, the offspring solutions were created by means of the crossover operator, which selects randomly subexpressions (corresponding to subtrees in the graphical representation in Fig. 1) in the two parent solutions and exchanges them. Then, for a small (.05) fraction of the population, the mutation operator randomly selects a subexpression and replaces it by other subexpression generated at random. Both these genetic operators obey the so-called *strong typing* principle [15], i.e. they yield individuals correct with respect to the GPVIS syntax.

Special precautions have been undertaken to prevent overfitting of hypotheses to the training data. This issue is of special importance, as the individuals in genetic programming usually tend to grow in an unlimited way, because large expressions are more resistant to destruction of performance in recombination process. The fitness function was extended by an additional penalty term implementing the so-called *parsimony pressure*. Solutions growing over 100 terms (nodes of expression tree) were linearly penalized with the evaluation decreasing to 0 when the threshold of 200 terms was reached. In pairwise comparison used in GPPO, solution growing over 100 terms is always outranked, no matter how well it performs on the training set.

After evolving the classifiers for particular class pairs, the binary classifiers (GPVIS expressions) had been combined to form the n^2 classifier, which was then tested on an independent test set. The test set contained 2000 instances, i.e. 200 images for each of 10 decision classes, selected randomly from the testing part of the MNIST database (containing digits written by different people as in the case of training set [21]).

5.2.4 Presentation of Results

Table 1 presents the comparison of the pattern recognition programs formulated in GPVIS obtained in GP and GPPO runs. Although finally the most important outcome is the accuracy of classification on the test set for the entire 10-class problem, the table contains also in part the results concerning binary classification problems. That gives us more statistical insight into the results. Particular rows describe experiments with different maximal number of generations as stopping condition. Due to the use of metaclassifiers, each row summarizes the results of 45 pairs of experiments, each consisting of GP and GPPO search starting from the same initial population. The table includes:

- the maximal number of generations allowed for the run ('Max. # of generations'),
- the number of pairs of GP and GPPO runs (per total of 45) for which the best solution evolved in GPPO yielded strictly better fitness (accuracy of classification on the training set) than the best one obtained from 'plain' GP ('GPPO better'),

- the average increase of accuracy of classification on the training set obtained by GPPO in comparison to GP ('Avg. inc. of acc.'),
- the accuracy of classification of the compound n^2 metaclassifier on the training set and test set for GP and GPPO ('Metaclassifier accuracy'),
- the size of the metaclassifier, measured as the number of terms of GPVIS expression ('Classifier size').

Table 1. Comparison of the pattern recognition programs evolved in GP and GPPO runs (detailed description in text); accuracy of classification expressed in percents

Max. # of genera-tions	Training set				Test set		Classifier size (# of terms)	
	GPPO better	Avg. inc. of acc.	Metaclassifier accuracy		Metaclassifier accuracy			
			GP	GPPO	GP	GPPO	GP	GPPO
20	22/45	-0.31	**55.9**	55.6	46.0	**47.5**	2639	2583
40	27/45	0.43	59.4	**62.4**	51.4	**54.0**	2916	2812
60	31/45	1.29	62.8	**65.3**	**54.6**	53.5	3068	2939
80	36/45	2.34	62.6	**66.1**	55.0	**57.7**	2988	2902
100	36/45	2.55	62.4	**66.7**	54.9	**58.4**	3042	3109

6 Conclusions and Future Research Directions

As far as the binary classification problems are concerned, GPPO reaches on average better solution than those obtained by means of GP w.r.t. the performance on the training set (except for the case when the maximum number of generations was set to 20). Each positive increase shown in column 3 of Table 1 is statistically significant at 0.1 level with respect to the Wilcoxon's matched pairs signed rank test, computed for the results obtained by particular binary classifiers. The improvements seem to be attractive, bearing in mind the complexity of the visual learning task in the direct approach (see Section 5.1). It is also very encouraging that the difference in accuracy of classification grows as the evolution proceeds, what allows us to suppose that continuing the experiment would lead to even more convincing results.

To some extent, analogous conclusions may be drawn from the results concerning metaclassifiers (see columns 4-7 of Table 1). Except for run length 20 (first row of the table), GPPO metaclassifiers are superior on the training set. On the test set, that superiority is also observable, except for the experiment with generations limit set to 60, where GP wins the competition (rather accidentally). Note also that the GPPO solutions have similar size to those computed by GP (see columns 8-9 of Table 1).

It should be stressed that, due to limits on available computer resources, these encouraging improvements have been obtained with relatively small populations (50 individuals), restricted set of fitness cases (100 images for each binary problem), and short run lengths (20 - 100). That is why the absolute values of accuracy of classification reached by both GP and GPPO algorithms are rather not impressive in comparison to the 'handcrafted' methods or, for instance, neural networks [21].

However, the aim of this study was to draw a comparison and to check the usefulness of hypothesis evaluation by means of binary relation. We plan to carry out separate series of experiments devoted to the maximization of the accuracy of classification of the compound metaclassifier on the test set. With larger populations and longer runs better results are expected.

The general qualitative result obtained in the experiment is that evolutionary search involving pairwise comparison of solutions (GPPO) outperforms the 'plain' genetic programming (GP) on average. Thus, it seems to be worthwhile to control the search of the hypothesis space by means of an incomparability-allowing, pairwise comparison relation. Such an evaluation method protects the novel solutions from being discarded in the search process, even if they exhibit minor fitness in scalar terms. In other words, in the presence of an order, we do not necessarily have to look for the mediation of numbers. Formulating the reasons for GPPO superiority in other terms, GPPO benefits from the more detailed information concerning the 'behavior' of particular solutions on the training set.

Further work on this topic may concern different issues. In particular, we are still looking for other definitions of outranking than those discussed in this paper, especially for the parameter-free ones. In our opinion it would be also useful to make the selection procedure presented in Section 4 more elegant. And, last but not least, as the proposed framework is rather general and offers an easy possibility of adapting to other environment, we consider its application in different pattern analysis problems, like object detection in outdoor images.

Acknowledgements

The author would like to thank Yann LeCun for making available the MNIST database. This work was supported from the KBN research grant no. 8T11F 006 19.

References

1. Bala, J.W., De Jong, K.A., Pachowicz, P.W.: Multistrategy learning from engineering data by integrating inductive generalization and genetic algorithms. In: Michalski, R.S. , Tecuci, G. (eds.): Machine learning. A multistrategy approach. Volume IV. Morgan Kaufmann, San Francisco (1994) 471–487
2. Beasley, D., Bull, D.R., Martin, R.R.: A Sequential Niche Technique for Multimodal Function Optimization. Evolutionary Computation 1 (2), (1993) 101–125
3. Chan, P.K., Stolfo, S.J.: Experiments on multistrategy learning by meta-learning. Proceedings of the Second International Conference on Information and Knowledge Management (1993)
4. De Jong, K.A.: An analysis of the behavior of a class of genetic adaptive systems. Doctoral dissertation, University of Michigan, Ann Arbor (1975)
5. De Jong, K.A., Spears, W.M., Gordon, D.F.: Using genetic algorithms for concept learning. Machine Learning, 13 (1993) 161–188
6. Dubois, D., Prade, H.: Fuzzy sets and systems. theory and applications. Academic Press, New York (1980)

7. Goldberg, D.: Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading (1989)
8. Goldberg, D.E., Deb, K., Korb, B.: Do not worry, be messy. Proceedings of the Fourth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo (1991) 24–30
9. Goldberg, D., Richardson, J.: Genetic algorithms with sharing for multimodal function optimization. Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms (1987) 41–49
10. Gonzalez, R.C., Woods, R.E.: Digital image processing. Addison-Wesley, Reading (1992)
11. Harik, G.: Finding multimodal solutions using restricted tournament selection. In: Eshelman, L. J. (ed.): Proceedings of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, San Francisco (1995) 24–31
12. Holland, J.H.: Adaptation in natural and artificial systems. University of Michigan Press, Ann Arbor (1975)
13. Jelonek, J., Stefanowski, J.: Experiments on solving multiclass learning problems by n^2-classifier. In: Nedellec, C., Rouveirol, C. (eds.): Lecture Notes in Artificial Intelligence, Vol. 1398. Springer-Verlag, Berlin Heidelberg New York (1998) 172–177
14. Johnson, M.P.: Evolving visual routines. Master's Thesis, Massachusetts Institute of Technology (1995)
15. Koza, J.R.: Genetic programming - 2. MIT Press, Cambridge (1994)
16. Koza, J.R., Keane, M., Yu, J., Forrest, H.B., Mydlowiec, W.: Automatic Creation of Human-Competetive Programs and Controllers by Means of Genetic Programming. Genetic Programming and Evolvable Machines 1 (2000) 121–164
17. Krawiec, K.: Constructive induction in picture-based decision support. Doctoral dissertation, Institute of Computing Science, Poznań University of Technology, Poznań (2000)
18. Krawiec, K.: Constructive induction in learning of image representation. Research Report RA-006, Institute of Computing Science, Poznan University of Technology (2000)
19. Krawiec, K.: Pairwise Comparison of Hypotheses in Evolutionary Learning. Proceedings of The Eighteenth International Conference on Machine Learning (2001)
20. Langley, P. Elements of machine learning. San Francisco: Morgan Kaufmann (1996)
21. LeCun, Y., Jackel, L. D., Bottou, L., Brunot, A., et al.: Comparison of learning algorithms for handwritten digit recognition. International Conference on Artificial Neural Networks (1995) 53–60
22. Mahfoud, S.W.: A Comparison of Parallel and Sequential Niching Methods. In: Eshelman, L.J. (ed.): Proceedings of the Sixth International Conference on Genetic Algorithms. Morgan Kaufmann, San Mateo (1995) 136–143
23. Mitchell, T.M.: An introduction to genetic algorithms. MIT Press, Cambridge (1996)
24. Mitchell, T.M.: Machine learning. McGraw-Hill, New York (1997)
25. Poli, R. Genetic programming for image analysis, (Technical Report CSRP-96-1). The University of Birmingham (1996)
26. Rissanen, J.: A universal prior for integers and estimation by minimum description length. The Annals of Statistics, 11 (1983) 416–431
27. Sanchez, E.: Inverses of fuzzy relations. Application to possibility distributions and medical diagnosis. Proc. IEEE Conf. Decision Control, New Orleans 2, (1979) 1384–1389
28. Schaffer, J.D.: Multiple objective optimization with vector evaluated genetic algorithms. Proceedings of the First International Conference on Genetic Algorithms and their Applications. Lawrence Erlbaum Associates, Hillsdale (1985)
29. Teller, A., Veloso, M.: A controlled experiment: evolution for learning difficult image classification. Lecture Notes in Computer Science, Vol. 990. Springer-Verlag, Berlin Heidelberg New York (1995) 165–185
30. Vafaie, H., Imam, I.F.: Feature selection methods: genetic algorithms vs. greedy-like search. Proceedings of International Conference on Fuzzy and Intelligent Control Systems (1994)

31. Van Veldhuizen, D.A.: Multiobjective evolutionary algorithms: classifications, analyses, and new innovations. Doctoral dissertation, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Ohio (1999)
32. Vincke, P.: Multicriteria decision-aid. John Wiley & Sons, New York (1992)
33. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. In: Motoda, H., Liu, H. (eds.), Feature extraction, construction, and subset selection: A data mining perspective. Kluwer Academic, New York (1998)

Featureless Pattern Recognition in an Imaginary Hilbert Space and Its Application to Protein Fold Classification

Vadim Mottl[1], Sergey Dvoenko[1], Oleg Seredin[1],
Casimir Kulikowski[2], and Ilya Muchnik[2]

[1] Tula State University, Lenin Ave. 92, 300600 Tula, Russia
mottl@atm.tsu.tula.ru
[2] Rutgers University, P.O. Box 8018, Piscataway, NJ 08855, USA
kulikows@cs.rutgers.edu

Abstract. The featureless pattern recognition methodology based on measuring some numerical characteristics of similarity between pairs of entities is applied to the problem of protein fold classification. In computational biology, a commonly adopted way of measuring the likelihood that two proteins have the same evolutionary origin is calculating the so-called alignment score between two amino acid sequences that shows properties of inner product rather than those of a similarity measure. Therefore, in solving the problem of determining the membership of a protein given by its amino acid sequence (primary structure) in one of preset fold classes (spatial structure), we treat the set of all feasible amino acid sequences as a subset of isolated points in an imaginary space in which the linear operations and inner product are defined in an arbitrary unknown manner, but without any conjecture on the dimension, i.e. as a Hilbert space.

1 Introduction

The classical pattern recognition theory deals with objects represented in a finite-dimensional space of their features that are assumed to be defined in advance, before real objects subject to classification are observed. The emphasis on the feature-based representation of objects is reflected in the name of the most popular method of machine learning for pattern recognition called the support vector method [1,2].

At the same time, there exists a wide class of applications in which it is easy to evaluate some numerical characteristics of pairwise relationship between any two objects, but it is hard to indicate a set of rational individual attributes of objects that could form the axis of a feature space.

As an alternative to the feature-based methodology, R. Duin and his colleagues [3,4,5] proposed a featureless approach to pattern recognition, in which objects are assumed to be represented by appropriate measures of their pairwise similarity or dissimilarity. It is just this idea we use here as a basis for creating techniques of protein fold class recognition, i.e. allocating a protein, given by the primary chemical structure of its polymerous molecule as a sequence of amino acids (to be exact, their residues) from the alphabet of 20 amino acids existing in nature, over a finite set of typical spatial structures, each associated with a specific manner in which the primary amino acid chains fold in space under a highly complicated combination of numerous

This work is supported, in part, by the Russian Foundation of Basic Research, Grant No. 99-01-00372, and the State Scientific Program of the Russian Federation "Promising Information Technologies".

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 322–336, 2001.

physical forces [6,7]. We lean here upon the compactness hypothesis that is understood as the tendency of proteins with "similar" amino acid chains to belong to the same fold class [8].

It is common practice in computational biology to measure the proximity between two amino acid chains as the logarithmic likelihood ratio of two hypotheses, the main hypothesis that both of them originate from the same unknown protein as result of independent successions of local evolutionary mutations versus the null hypothesis that the chains are completely occasional combinations over the alphabet of 20 amino acids [9]. The generally accepted way of measuring such a likelihood ratio is calculating the so-called alignment score between two amino acid sequences, which is based on finding an appropriate consensus sequence from which both sequences might be obtained as result of as a small number of local corrections as possible, namely, deletions, insertions and substitutions of single amino acids [10,11].

By its nature, the logarithmic likelihood ratio may take as positive as well negative values. In addition, such a ratio calculated for an amino acid sequence with itself gives different values for different proteins. As a result, it is hard to interpret the pairwise alignment score as a similarity measure. In this work, we pose the heuristic hypothesis that the set of all feasible amino acid sequences may be considered as a subset of isolated points in an imaginary Hilbert space in which the linear operations are defined in an arbitrary unknown manner, and the role of inner product is played by the alignment score between the respective pair of amino acid chains.

Such an assumption allows for treating the sought-for decision rule of pattern recognition by the principle "one class against another one" as a discriminant hyperplane immediately in the Hilbert space of objects. However, the absence of coordinate axes prevents from finding the "direction element" of the hyperplane, i.e. an element of the Hilbert space that splits all the space points into two nonintersecting regions by values of scalar products with it.

Therefore, we propose to use an assembly of selected "representative" objects as a basis in the Hilbert space of all the feasible objects. The elements of the basic assembly are not assumed to be classified, their mission is to serve as coordinate axes of a finite-dimensional subspace, onto which any new object, including those forming the classified training sample, could be projected by calculating inner products with the basic elements.

The idea of making distinction between the unclassified basic assembly and classified training sample appears to be quite reasonable for the problem of protein fold class recognition, because the number of proteins whose spatial structure is known is much less than the number of proteins with known amino acid chains.

2 The Problem of Protein Fold Class Recognition

The problem of finding the spatial structure of a protein represented by its primary amino acid sequence is a challenge posed by the nature. On the one hand, the necessity of such algorithms is dictated by the fact that application of usual physical techniques of magnetic resonance and X-ray analysis is problematic in most cases. Although the number of proteins whose spatial structure is known ever grows, the gap between the number of known amino acid sequences and that of known spatial structures is increasing dramatically. On the other hand, the "existence theorem" is proved by nature itself, because it has been never observed that an amino acid chain had more than one spatial structure.

Each protein has its specific spatial organization which does not coincide with that of any other protein. The main principle of establishing the spatial structure of a given

protein from its amino acid chain consists in finding, for the given chain, the most appropriate structure from a bank of known structures and their fragments. For each amino acid residue in the chain forming a protein of a known structure, the vector of some quantitative features is evaluated which are assumed to be responsible for the spatial position of this residue in the three-dimensional structure. The succession of such features along the amino acid chain is called the profile of this structure. The same features are evaluated for the amino acid chain of the new protein, whereupon the succession obtained is compared with profiles of known structures by alignment of positions in this succession and in the respective profile with respect to eventual insertions and deletions. Such a principle named threading [6] is fraught with enumeration of a large number of known structures.

Despite the uniqueness of the spatial structure of each protein, it is the usual case that large groups of evolutionary allied proteins have very similar spatial structures. In this sense, there exist "much less" spatial structures than primary ones. Of course, the classification of spatial structures is a problem which is not simple, but once a version of classification is accepted, the problem of assigning an amino acid chain to a class of spatial structures falls into the competence area of pattern recognition.

In an earlier series of experiments [7], an attempt was made to describe the primary amino acid sequence of a protein by vector of its numerical features and consider it as a point in the respective linear vector space. In particular, the primary structure of a protein was represented by frequencies with which amino acids of the polar, neutral and hydrophobic type and their pairs occur in it.

The results of those experiments cannot be assessed as quite successful, to all appearance, because of an immensely rich actual diversity of amino acid properties that may play an important part in forming the spatial structure of a protein. Therefore, we turn here to the featureless formulation of the fold class recognition problem.

When studying the structure and properties of proteins, one of commonly used instruments is the characteristic of mutual similarity of two amino acid sequences $\omega' = (a_1,...,a_N)$ and $\omega'' = (b_1,...,b_K)$ given by an appropriate pair-wise alignment procedure (Fig.1). Procedures of such a kind lean upon a preset similarity matrix for all 210 pairs of 20 amino acids. Such matrices are called substitution matrices and characterize each amino acid pair (a,b) by logarithmic ratio of, first, the probability of their independent occurrence in two amino acid chains p_{ab} as result of evolutionary substituting the same unknown amino acid c in a common ancestor chain, and, second, the product of general probabilities q_a and q_b of their occurrence in arbitrary sequences [9]:

$$s(a,b) = \log(p_{ab}/q_a q_b). \qquad (1)$$

The log likelihood ratio $s(a,b)$ is positive if the probability that these two amino acids have a common ancestor is greater than the product of their general probabilities, equals zero in the indifferent case, and is negative if the hypothesis of their common origin is less likely than that of the null hypothesis of their independent occasional appearance.

```
ω′ : TNPGNASSTTTTKPTTTS-----RGLKTINETDPCIKNDSCTG
ω″ : GS----ATSTPATSTTAGTKLPCVRNKTDSNLQSCNDTIIEKE
 i = 12    34567  ···
```

Fig. 1. Fragment of an aligned pair of amino acid chains from the protein family *Envelope glycoprotein GP120* in the database Pfam.

There are several versions of substitution matrices [9,12,13], but each of them is result of observations in large sets of proteins aligned in that or other manner by experienced biologists in accordance with their intuition based, in its turn, on that or other model of evolution.

The numerical measure of the proximity of two proteins represented by their amino acid chains is determined as the greatest possible sum of $s(a_{j_i}, b_{k_i})$ over all related pairs of amino acids (j_i, k_i), $i = 1,2,3,...$, in a pair-wise alignment with respect to some penalties posed on the presence and length of gaps (Fig. **Fehler! Verweisquelle konnte nicht gefunden werden.**):

$$\mu(\omega', \omega'') = \sum_i s(a_{j_i}, b_{k_i}) - (\text{gap length penalties}). \quad (2)$$

In our experiments we used this similarity measure of amino acid chains measured by the commonly adopted alignment procedure Fasta 3 [10,11] with substitution matrix Blossum 50 [9].

As the set of experimental data, we took the collection of proteins selected by Dr. Sun-Ho Kim from Lawrence Berkley National Laboratory in the USA. The collection contains 396 protein domains, i.e. relatively isolated fragments of amino acid chains, chosen from the SCOP Database (Structural Classification of Proteins). The protein domains forming the collection belong to 51 fold classes listed in Table 1. The principle of selection was to provide a low similarity of amino acid sequences within each family, with which purpose only those protein domains were chosen whose similarity (2) to other selected domains did not exceed a preset threshold. Such a principle of selection resulted in protein domain families of different size.

3 The Pair-wise Alignment Score of Two Amino Acid Chains as Their Inner Product in an Imaginary Hilbert Space

It appears natural to interpret the log likelihood ratio for two amino acids $s(a,b)$ (1) as experimentally registered outward exhibition of the actual proximity of their hidden properties. Let these properties be expressed by some hidden vectors $\mathbf{y}_a$ and $\mathbf{y}_b$ for which the notion of inner product is defined $(\mathbf{y}_a, \mathbf{y}_b)$, then the structure of (1) suggests the idea to consider $s(a,b)$ as a rough measure of it: $s(a,b) \cong (\mathbf{y}_a, \mathbf{y}_b)$.

By analogy to a single summand, the score of the alignment as a whole (2) may also be interpreted as inner product of the respective combined feature vectors of two proteins $\mu(\omega', \omega'') \cong (\mathbf{x}_{\omega'}, \mathbf{x}_{\omega''})$ in an imaginary linear feature space. The greater the positive value of the similarity, the more "synchronous" are some essential properties of amino acids along the polypeptide chain, the zero value says about full lack of agreement what corresponds to the notion of orthogonality, and a negative value should be interpreted as "opposite phases" of amino acid properties along the chains.

However, this is not more than a cursory analogy. For an accurate justification of the hypothesis that there exists a Hilbert space in which the set of proteins could be embedded, we should show that the score matrix of any finite assembly of proteins tends to be nonnegative definite or, at least, can be approximated by such a matrix.

We checked this hypothesis for an assembly of 396 proteins (Table 1) by way of calculating all the eigenvalues of the score matrix obtained by the pair-wise alignment procedure Fasta 3 [10,11] with the substitution matrix Blossum 50 [9]. All the eigenvalues turned out to be positive (Fig. 2).

Table 1. Dr. Kim's collection of proteins.

	Fold class	Size		Fold class	Size
1	Globin	12	27	Flavodoxin	9
2	Cytochrome C	7	28	Adenine nucleotide alpha hydroclase	4
3	Four-helical bundle	8	29	Rossmann-fold domains	14
4	Ferritin	8	30	Thiamin-binding	3
5	4-gelical cytokines	11	31	P-loop containing NTP hydrolases	9
6	EF Hand	13	32	Thioredoxin fold	9
7	Cyclin	4	33	Restriction endonucleases	5
8	Cytochrome P450	5	34	Ribonuclease H motif	9
9	Immunoglobin beta – sandwich	31	35	Phosphoribosyltransferases (PRTases)	3
10	Common fold of difteria toxin / transcription factors / cytochrome	5	36	S-adenosyl-L-methionine-dependent methyltransferases	5
11	Cupredoxins	9	37	Alpha / beta-Hydrolases	12
12	C2 domain	3	38	Phosphorylase / hydrolase	5
13	Viral coat and capsid proteins	15	39	Periplastic binding protein I	7
14	Crystallins / protein S / yeast killer toxin	5	40	Periplastic binding protein II	7
15	Galastose-binding domain	4	41	Lysozyme	4
16	ConA lectins / glucanases	8	42	Cysteine proteinases	4
17	OB-fold	17	43	Beta-Grasp	8
18	Beta-Trefoil	5	44	Cystatin	7
19	Reductase / isomerase / elongation factor	4	45	Ferredoxin	20
20	Trypsin serine proteases	6	46	Zincin	7
21	Acid proteases	5	47	N-terminal nucleophile aminohydrolases	4
22	PH domain	7	48	ADP-ribosylation	4
23	Lipocalings	6	49	C-type lectin	6
24	Double-stranded beta-helix	6	50	Protein kinases (PK), catalytic core	4
25	Barrel-sandwich hybrid	6	51	Beta-Lactamase / D-ala carboxypeptidase	3
26	TIM-barrel	28			

The conclusion suggests itself that the pair-wise similarity measure determined by the procedure Fasta 3 possesses properties having much in common with those of inner product. This circumstance should be considered as a reason in favor of the theoretical applicability of the principle of featureless pattern recognition in a Hilbert space to the problem of protein fold class recognition.

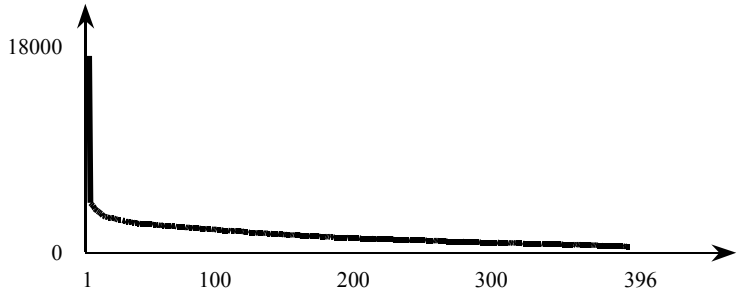


Fig. 2. Eigenvalues of Dr.Kim's collection of proteins: $\lambda_{max} = 16621$, $\lambda_{min} = 304$; all eigenvalues are positive.

4 Hilbert Space of Classified Objects and Optimal Discriminant Hyperplane

Let the set Ω of all feasible objects under consideration $\omega \in \Omega$ is partitioned into two classes $\Omega_1 = \{\omega \in \Omega : g(\omega) = 1\}$ and $\Omega_{-1} = \{\omega \in \Omega : g(\omega) = -1\}$ by an unknown indicator function $g(\omega) = \pm 1$. The main idea of the featureless approach to pattern recognition consists in treating the set Ω as a Hilbert space in which the linear operations and inner product are defined in an arbitrary manner under the usual constraints:

(1) addition is symmetric and associative $\omega' + \omega'' = \omega'' + \omega' \in \Omega$, $\omega' + (\omega'' + \omega''') = (\omega' + \omega'') + \omega'''$;
(2) there exists an origin $\phi \in \Omega$ such that $\omega + \phi = \omega$ for any element $\omega \in \Omega$;
(3) there exists the inverse elements $(-\omega) + \omega = \phi$ for any $\omega \in \Omega$;
(4) multiplication by a real coefficient $c\,\omega \in \Omega$, $c \in \mathsf{R}$, is associative $(cd)\omega = c(d\,\omega)$ and $1\omega = \omega$ for any $\omega \in \Omega$;
(5) addition and multiplication are distributive $c(\omega' + \omega'') = c\,\omega' + c\,\omega''$, $(c + d)\omega = c\,\omega + d\,\omega$;
(6) inner product of elements is symmetric $(\omega', \omega'') = (\omega'', \omega') \in \mathsf{R}$ and linear $(\omega, \omega' + \omega'') = (\omega, \omega') + (\omega, \omega'')$, $(\omega, c\,\omega') = c\,(\omega, \omega')$;
(7) inner product of an element with itself possesses the properties $(\omega, \omega) \geq 0$, $(\omega, \omega) = 0$ if and only if $\omega = \phi$ and gives the norm $\|\omega\| = (\omega, \omega)^{1/2} \geq 0$.

It is not meant that all the elements of the Hilbert space Ω do exist in reality. We consider really existing objects as making a subset $\widetilde{\Omega}$ of isolated points in Ω, whereas all the remaining elements are nothing else than products of our imagination.

It is just the extension of $\widetilde{\Omega}$ to Ω what allows speaking about "sums" of really existing objects and their "products" with real-valued coefficients.

It is assumed that even if an element of the Hilbert space $\omega \in \Omega$ really exists $\omega \in \widetilde{\Omega} \subset \Omega$, it cannot be perceived by the observer in any other way than through its inner products (ω, ω') with other really existing elements $\omega' \in \widetilde{\Omega} \subset \Omega$. If $\vartheta \in \Omega$ is a fixed element of the Hilbert space, an imaginary one in the general case, the real-valued linear discriminant function $d(\omega \,|\, \vartheta, b) = (\vartheta, \omega) + b$, where $b \in \mathsf{R}$ is a constant, may be used as decision rule $\hat{g}(\omega) : \Omega \to \{1, -1\}$ of judging on the hidden class-membership of an arbitrary object $\omega \in \Omega$, might it really exist or not:

$$d(\omega \,|\, \vartheta, b) = (\vartheta, \omega) + b \begin{cases} > 0 \to \hat{g}(\omega) = 1, \\ < 0 \to \hat{g}(\omega) = -1. \end{cases} \tag{3}$$

Here the element $\vartheta \in \Omega$ plays the role of the direction element of the respective discriminant hyperplane in the Hilbert space $(\vartheta, \omega) + b = 0$.

However, we have, so far, no constructive instrument of choosing the direction element $\vartheta \in \Omega$ and, hence, the decision rule of recognition, because, just as any element of Ω, it can be defined only by its inner products with some other fixed elements that exist in reality.

Let the observer have chosen an assembly of really existing objects $\Omega^0 = \{\omega_1^0, ..., \omega_n^0\} \subset \Omega$, called the basic assembly, which is not assumed to be classified, in the general case, and, therefore, it is not yet a training sample. The basic assembly will play the role of a finite basis in the Hilbert space that defines an n-dimensional subspace

$$\Omega_n(\omega_1^0, ..., \omega_n^0) = \left\{ \omega \in \Omega : \omega = \sum\nolimits_{i=1}^{n} a_i \omega_i^0 \right\} \subset \Omega. \tag{4}$$

We restrict our consideration to only those discriminant hyperplanes whose direction elements belong to $\Omega_n(\omega_1^0, ..., \omega_n^0)$, i.e. can be expressed as linear combinations

$$\vartheta(\mathbf{a}) = \sum_{i=1}^{n} a_i \omega_i^0, \quad \mathbf{a} \in \mathsf{R}^n. \tag{5}$$

The respective parametric family of discriminant hyperplanes $(\vartheta(\mathbf{a}), \omega) + b = \sum_{i=1}^{n} a_i (\omega_i^0, \omega) + b = 0$ and, so, linear decision rules

$$d(\omega \,|\, \vartheta(\mathbf{a}), b) = \sum_{i=1}^{n} a_i (\omega_i^0, \omega) + b \begin{cases} > 0 \to \hat{g}(\omega) = 1, \\ < 0 \to \hat{g}(\omega) = -1, \end{cases} \quad \omega \in \Omega, \tag{6}$$

will be completely defined by inner products of elements of the Hilbert space with elements of the basic assembly (ω_i^0, ω), $i = 1, ..., n$. We shall consider the totality of these values for an arbitrary element $\omega \in \Omega$ as its real-valued "feature vector"

$$\mathbf{x}(\omega) = \left(x_1(\omega) \cdots x_n(\omega)\right)^T \in \mathsf{R}^n, \quad x_i(\omega) = (\omega_i^0, \omega). \tag{7}$$

Mark that if $(\vartheta(\mathbf{a}), \omega) = 0$ then $(\omega_i^0, \omega) = 0$ for all $\omega_i^0 \in \Omega^0$. This means that by choosing the direction elements in accordance with (5) we restrict our consideration to only those discriminant hyperplanes which are orthogonal to the subspace spanned over the basic assembly of objects. As a result, all elements of the Hilbert space that have the same inner products with basic elements $\mathbf{x} = \left((\omega_1^0, \omega) \cdots (\omega_n^0, \omega)\right)^T$, or, in other

words, the same projection on the basic subspace $\Omega_n(\omega_1^0,...,\omega_n^0)$ (4), will be assigned the same class $\hat{g}(\omega)=\pm 1$ by linear decision rules (6). Therefore, we call the features (7) projectional features of Hilbert space elements.

We have come to a parametric family of decision rules of pattern recognition in a Hilbert space (6) that lean upon projectional features of objects:

$$d\big(\mathbf{x}(\omega)\,|\,\mathbf{a},b\big)=\mathbf{a}^T\mathbf{x}(\omega)+b\begin{cases}>0\to\hat{g}(\omega)=1,\\<0\to\hat{g}(\omega)=-1,\end{cases}\qquad \omega\in\Omega. \tag{8}$$

Thus, the notion of projectional features reduces, at least, superficially, the problem of featureless pattern recognition in a Hilbert space to the classical problem of pattern recognition in a usual linear space of real-valued features.

Let the observer be submitted a classified training sample of objects $\Omega^*=\{\omega_1,...,\omega_N\}\subset\Omega$, $g_1=g(\omega_1),...,g_N=g(\omega_N)$, that does not coincide, in the general case, with the basic assembly $\Omega^0=\{\omega_1^0,...,\omega_n^0\}$. The observer has no other way of perceiving them than to calculate their inner products with objects of the basic assembly, what is equivalent to evaluating their projectional features

$$\mathbf{x}(\omega_j)=\big(x_1(\omega_j)\cdots x_n(\omega_j)\big)^T=\big((\omega_1^0,\omega_j)\cdots(\omega_n^0,\omega_j)\big)^T\in\mathsf{R}^n.$$

Parameters of the discriminant hyperplane $\mathbf{a}\in\mathsf{R}^n$ and $b\in\mathsf{R}$ (8) should be chosen so that the training objects would be classified correctly with a positive margin $\xi>0$:

$$d\big(\omega_j\,|\,\vartheta(\mathbf{a}),b\big)=\sum_{i=1}^{n}a_i(\omega_i^0,\omega_j)+b=\mathbf{a}^T\mathbf{x}(\omega_j)+b\begin{cases}\geq\xi\ \text{ when }\ g(\omega_j)=1,\\\leq-\xi\ \text{ when }\ g(\omega_j)=-1.\end{cases} \tag{9}$$

If the training sample is linearly separable with respect to the basic assembly, there exists a family of hyperplanes that satisfy these conditions. It is clear that the margin ξ remains positive after multiplying the pair $\big(\vartheta(\mathbf{a})\in\Omega,b\in\mathsf{R}\big)$ with a positive coefficient $\big(c\,\vartheta(\mathbf{a})\in\Omega,c\,b\in\mathsf{R}\big)$, $c>0$, thus, it is sufficient to consider direction elements of a preset norm $\|\vartheta(\mathbf{a})\|=\big(\vartheta(\mathbf{a}),\vartheta(\mathbf{a})\big)^{1/2}=\text{const}$. One of them, for which $\xi\to\max$ and the conditions (9) are met, will be called the optimal discriminant hyperplane in the Hilbert space.

Because the direction element of the discriminant hyperplane is determined here by a finite-dimensional parameter vector, such a problem, if considered in the basic subspace $\Omega_n(\omega_1^0,...,\omega_n^0)$ (4), completely coincides with the classical statement of the pattern recognition problem as that of finding the optimal discriminant hyperplane. The same reasoning as in [2] leads to the conclusion that the maximum margin is provided by choosing the direction element $\vartheta(\mathbf{a})\in\Omega$ and threshold $b\in\mathsf{R}$ from the condition

$$\|\vartheta(\mathbf{a})\|^2\to\min,\qquad g_j\big[\big(\vartheta(\mathbf{a}),\omega_j\big)+b\big]\geq 1,\qquad j=1,...,N. \tag{10}$$

However, such an approach becomes senseless in case the classes are inseparable in the basic subspace, and the constraints (9) and, hence, (10) are incompatible. To design an analogous criterion for such training samples, we, just as V. Vapnik, admit nonnegative defects $g_j\big[\big(\vartheta(\mathbf{a}),\omega_j\big)+b\big]\geq 1-\delta_j$, $\delta_j\geq 0$, and use a compromise criterion $(\vartheta,\vartheta)+C\sum_{j=1}^{N}\delta_j\to\min$ with a sufficiently large positive coefficient C meant to give preference to the minimization of these defects. So, we come to the

following formulation of the generalized problem of finding the optimal discriminant hyperplane in the Hilbert space that covers both the separable and inseparable case:

$$\begin{cases} \| \vartheta(\mathbf{a}) \|^2 + C\sum_{j=1}^{N} \delta_j \to \min, \\ g_j\left(\mathbf{a}^T \mathbf{x}(\omega_j) + b\right) \geq 1 - \delta_j, \quad \delta_j \geq 0, \quad j = 1,...,N. \end{cases} \tag{11}$$

5 Choice of the Norm of the Direction Element

The norm of the direction element of the sought-for hyperplane can be understood, at least, in two ways, namely whether as that of an element of the Hilbert space $\vartheta \in \Omega$ or as the norm of its parameter vector in the basic subspace $\mathbf{a} \in \mathsf{R}^n$. In the former case we have, in accordance with (5),

$$\| \vartheta(\mathbf{a}) \|^2 = \left(\vartheta(\mathbf{a}), \vartheta(\mathbf{a})\right)^2 = \sum_{i=1}^{n}\sum_{l=1}^{n} (\omega_i^0, \omega_l^0) a_i a_l = \mathbf{a}^T \mathbf{M} \mathbf{a}, \tag{12}$$

where $\mathbf{M} = \left((\omega_i^0, \omega_l^0), i, l = 1,...,n\right)$ is matrix $(n \times n)$ formed by inner products of basic elements $\omega_1^0,...,\omega_n^0$, whereas in the latter case

$$\| \vartheta(\mathbf{a}) \|^2 = \sum_{i=1}^{n} a_i^2 = \mathbf{a}^T \mathbf{a}. \tag{13}$$

In the "native" version of norm (12), the training criterion (11) is aimed at finding the shortest direction element $\vartheta \in \Omega$, and, so, all orientations of the discriminant hyperplane in the original Hilbert space are equally preferable. On the contrary, if the norm is measured as that of the vector of coefficients representing the direction element in the space of projectional features (13), the criterion (11) seeks the shortest vector $\mathbf{a} \in \mathsf{R}^n$ (13), so that equally preferable are all orientations of the hyperplane in R^n but not in Ω.

It is easy to see that if $\vartheta \in \Omega$ and $\omega \in \Omega$ are two arbitrary elements of a Hilbert space Ω, then the squared Euclidean distance from ω to its projection onto the beam formed by element ϑ equals $(\omega,\omega) - (\omega,\vartheta)^2 / (\vartheta,\vartheta)$. In its turn, it can be shown [8] that if $\mathbf{a}^T \mathbf{a} \to \min$ under the constraint $\left(\vartheta(\mathbf{a}), \vartheta(\mathbf{a})\right) = \mathbf{a}^T \mathbf{M} \mathbf{a} = \text{const}$, then $\sum_{j=1}^{n} \left(\omega_j, \vartheta(\mathbf{a})\right)^2 \to \max$, and, so, $\vartheta(\mathbf{a})$ tends to be close to the major inertia axis of the basic assembly.

Thus, training by criterion (11) with $\| \vartheta(\mathbf{a}) \|^2 = \mathbf{a}^T \mathbf{a}$, i.e. without any preferences in the space of projectional features, is equivalent to a pronounced preference in the original Hilbert space in favor of direction elements oriented along the major inertia axis of the basic assembly of object. As a result, the discriminant hyperplane in the Hilbert space tends to be orthogonal to that axis (Fig. 3).

This is out of significance if the region of major concentration of objects in the Hilbert space is equally stretched in all directions. But such indifference is rather an exclusion than a rule. It is natural to expect the distribution of objects be differently extended in different directions, what fact will be reflected by the form of the basic assembly and, then, by the training sample. In this case, a reliable decision rule of recognition exists only if objects of two classes are spaced just in one of the directions

where the extension is high. Therefore, it appears reasonable to escape discriminant hyperplanes oriented along the basic assembly even if the gap between the points of the first and the second class in the training sample has such an orientation, and prefer transversal hyperplanes (Fig. 3). It is just this preference that is expressed by the training criterion (11) with $\| \vartheta(\mathbf{a}) \|^2 = \mathbf{a}^T\mathbf{a} \to \min$ in contrast to $\| \vartheta(\mathbf{a}) \|^2 = \mathbf{a}^T\mathbf{M}\mathbf{a} \to \min$.

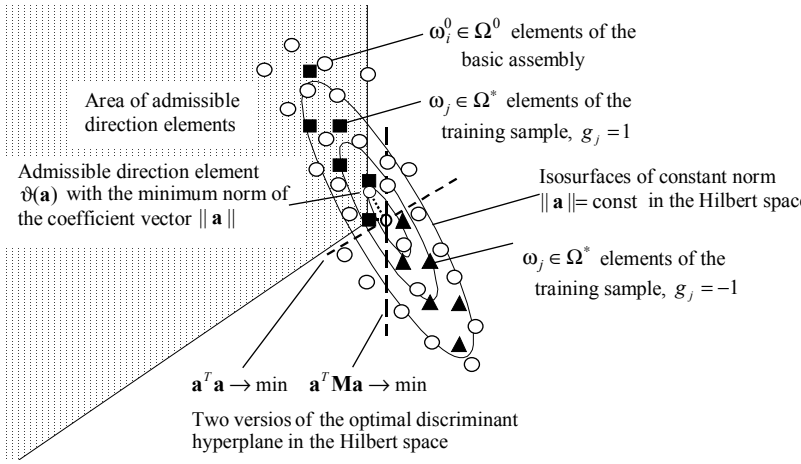


Fig. 3. Minimum norm of the direction vector of the discriminant hyperplane in the space of projectional features as criterion of training. In the original Hilbert space, the discriminant hyperplanes are preferred whose direction elements are oriented along the major inertia axis of the basic assembly.

6 Smoothness Principle of Regularization in the Space of Projectional Features

Actually, training by criterion $\mathbf{a}^T\mathbf{a} \to \min$ is nothing else than a regularization method that makes use of some information on the distribution of objects in the Hilbert space. This information is taken from the basic assembly and, so, should be considered as a priori one relative to the training sample. In case the distribution is almost degenerate in some directions, it is reasonable to prefer discriminant hyperplanes of transversal orientation even if the training sample suggests the longitudinal one as it is shown in Fig. 3.

In this Section, we consider another source of a priori information that may be drawn from the basic assembly of objects before processing the training sample. The respective regularization method follows from the very nature of projectional features, namely, from the suggestion that the closer are two objects of the basic assembly, the less should be the difference between the coefficients of their participating in the direction element of the discriminant hyperplane (5).

In the feature space of an arbitrary nature, there are no a priori preferences in favor of that or other mutual arrangement of classes, and the only source of information on the sought-for direction is the training sample. But in the space of projectional features

different directions are not equally probable, and it is just this fact that underlies the regularization principle considered here.

The elements of the projectional feature vector of an object $\omega \in \Omega$ are its scalar products with objects of the basic assembly $\mathbf{x}(\omega) = (x_1(\omega) \cdots x_n(\omega))^T \in \mathsf{R}^n$, $x_k(\omega) = (\omega, \omega_k^0)$, $\omega_k^0 \in \Omega^0 \subset \Omega$. The basic objects, in their turn, are considered as elements of the same linear Hilbert space and, so can be characterized by their mutual proximity. If two basic objects ω_j^0 and ω_k^0 are close to each other, the respective projectional features do not carry essentially different information on objects of recognition $\omega \in \Omega$, and it is reasonable to assume that the coefficients a_j and a_k in the linear decision rule should also take close values. Therein lies the a priori information on the direction vector of the discriminant hyperplane that is to be taken into account in the process of training.

In fact, the coefficients a_j are functions of basic points in the Hilbert space $a_j = a(\omega_j^0)$, and the regularization principle we have accepted consists in the a priori assumption that this function should be smooth enough. It is just this interpretation that impelled us to give such a principle of regularization the name of smoothness principle.

It remains only to decide how the pair-wise proximity of basic objects should be quantitatively measured. For instance, inner products $\mu_{jk} = (\omega_j, \omega_k)$ might be taken as such a measure. Then, the a priori information on the sought-for direction element can be easily introduced into the training criterion $\mathbf{a}^T\mathbf{a} + C\sum_{j=1}^{N}\delta_j \to \min$ in (11) with $\|\vartheta(\mathbf{a})\|^2 = \mathbf{a}^T\mathbf{a}$ as an additional quadratic penalty $\mathbf{a}^T(\mathbf{I} + \alpha\mathbf{B})\mathbf{a} + C\sum_{j=1}^{N}\delta_j \to \min$ where

$$\mathbf{a}^T\mathbf{B}\mathbf{a} = \frac{1}{2}\sum_{i=1}^{n}\sum_{l=1}^{n}\mu_{il}(a_i - a_l)^2, \quad \mathbf{B} = \begin{pmatrix} -\mu_{11} + \sum_{i=1}^{n}\mu_{1i} & \cdots & -\mu_{1n} \\ \vdots & \ddots & \vdots \\ -\mu_{n1} & \cdots & -\mu_{nn} + \sum_{i=1}^{n}\mu_{ni} \end{pmatrix},$$

and parameter $\alpha > 0$ presets the intensity of regularization.

Because the size of the training sample N is, as a rule, less than the dimensionality n of the space of projectional features, the subsamples of the first and the second class will most likely be linearly separable. On the force of this circumstance, when solving the quadratic programming problem (11) without regularizing penalty, the optimal shifts of objects will equal zero $\delta_j = 0$, $j = 1,...,N$. After introducing the regularization penalty, the errorless hyperplane may turn out to be unfavorable from the viewpoint of a priori preferences expressed by matrix $\mathbf{B}$ with sufficiently large coefficient α. In this case, the optimal hyperplane will sacrifice, if required, the correct classification of some especially nuisance objects of the training sample, what will result in positive values of their shifts $\delta_j > 0$.

7 Experiments on Protein Fold Class Recognition "One against One"

Experiments on fold class recognition were conducted with the collection of amino acid sequences of 396 protein domains grouped into 51 fold classes (Table 1). As the initial data set served the matrix 396×396 of pair-wise alignment scores obtained by alignment procedure Fasta 3 and considered as matrix of inner products of respective protein domains (ω_j, ω_k) in an imaginary Hilbert space.

In the series of experiments described in this Section, we solved the problem of pair-wise fold class recognition by the principle "one against one". There are $m = 51$ classes in the collection and, so, $m(m-1)/2 = 1275$ class pairs, for each of which we found a linear decision rule of recognition.

As the basic assembly $\Omega^0 = \{\omega_1^0, ..., \omega_n^0\}$, we took amino acid chains of 51 protein domains, $n = 51$, one from each fold class. As representatives of classes, their "centers" were chosen, i.e. the protein domains that gave the maximum sum of pair-wise scores with other members of the respective class. Thus, each protein domain was represented by a 51-dimensional vector of its projectional features (7).

For each of the 1275 class pairs, the training sample consisted of all protein domains making the respective two classes (Table 1). Thus, the size of the training sample varied from $N = 7$ for pairs of small classes, such as (50) *Protein kinases, catalic core* and (51) *Beta-Lactamase*, to $N = 59$ in two greatest classes (9) *Immunoglobin beta - sandwich* and (26) *TIM-barrel*.

We applied the technique of pattern recognition with preferred orientation of the discriminant hyperplane along the major inertia axis of the basic assembly in the Hilbert space. The quadratic programming problem (11) was solved for each of 1275 class pairs in its dual formulation [2].

A way of empirical estimating the quality of the decision rule immediately from the training sample offers the well-known leave-one-out procedure [2]. One of the objects of the full training sample containing N objects is left out at the stage of training, and the decision rule inferred from the remaining $N-1$ objects is applied to the left-out one. If the result of recognition coincides with the actual class given by the trainer, this fact is registered as success at the stage of examination, otherwise an error is fixed. Then the control object is returned to the training sample, another one is left out, and the experiment is run again. Such a procedure is applied to all the objects of the training sample, and the percentage of errors or correct decisions is calculated, which is considered as an estimate on the quality of the decision rule inferred from the full sample would it be applied to the general population.

In each of 1275 experiments, the separability of the respective two fold classes was estimated by such a procedure. Two rates were calculated for each class pair, namely, the percentage of correctly classified protein domains of the first and the second class. As the final estimate of the separability, the worst, i.e. the least, of these two percentages was taken.

As a result, the separability was found to be not worse than:

100% in 9% of all class pairs (completely separable class pairs),
90% in 14% of all class pairs,
80% in 32% of all class pairs,
70% in 53% of all class pairs.

The separability of 26 classes from more than one half of other classes is not worse than 70%. One class, namely, (50) *Protein kinases (PK), catalytic core*, showed its complete separability from all the classes.

On a data set of a lesser size, we checked how the pair-wise separability of fold classes will change if the number of basic proteins, i.e. the dimensionality of the projectional feature space, increases essentially. For this experiment, we took all the proteins of the collection as basic ones, so that the dimensionality of the projectional feature space became $n = 396$.

The same truncated data set was used for studying how the separability of classes is affected by normalization of the alignment scores between amino acid chains, what is equivalent to projection of respective points of the imaginary Hilbert space onto the unit sphere. If (ω', ω'') is inner product of two original points of the Hilbert space associated with the respective two protein domains ω' and ω'', the inner product of their projections $\overline{\omega}'$ and $\overline{\omega}''$ onto the unit sphere will be $(\overline{\omega}', \overline{\omega}'') = (\omega', \omega'') \Big/ \left(\sqrt{(\omega', \omega')} \sqrt{(\omega'', \omega'')} \right)$. We used these values, instead of (ω', ω''), as similarity measure of protein domain pairs for fold class recognition.

For this series of experiments, we selected 7 fold classes different by their size and averaged separability from other classes. The chosen classes that contain in sum 85 protein domains are shown in Table 2.

The results are presented in Table 3. As we see, the extension of the basic assembly improved the separability of the class pairs that participated in the experiment. As to the normalization of the alignment score, it led to an improvement with the small basic assembly and practically did not change the separability with the enlarged one.

Experimental study of effects of regularization was conducted with the same truncated data set (Table 2). We examined how the smoothness principle of regularization, expressed by the modified quadratic programming problem (4.1, improves the separability of fold classes “one against one” within the selected part of the collection. The separability of each of 21 pairs of classes was estimated by the leave-one-out procedure several times with different values of the regularization coefficient α. Each time, the separability of a class pair was measured by the worst percentage of correct decisions in the first and the second class, whereupon the averaged separability over all 21 class pairs was calculated for the current value of α.

Table 2. Seven fold classes selected for the additional series of experiments.

	Fold class	Size	Averaged separability from other classes
1	Globin	12	73.4 %
3	Four-helical bundle	8	70.8 %
4	Ferritin	8	60.4 %
5	4-gelical cytokines	11	66.0 %
10	Common fold of difteria toxin / transcription factors / cytochrome	5	65.2 %
12	C2 domain	3	8.2 %
26	TIM-barrel	28	52.6 %

Such a series of experiments was carried out twice, with original and normalized alignment scores. The dependence of the separability on the regularization coefficient in both series is shown in Fig. 4. In both series of experiments, a marked improvement of the separability is gained. The quality of training grows as the regularization coefficient increases, however, the improvement in not monotonic. A slight drop in separability with further increase in the coefficient after the maximum is attained arises from a too deep roughness of the decision rule adjustment.

Table 3. Averaged pair-wise separability of seven fold classes in four additional experiments.

Size of the basic assembly n	**Averaged separability**	
	Original score matrix (ω', ω'')	**Normalized score matrix** $(\overline{\omega}', \overline{\omega}'')$
51	63.5 %	69.4 %
396	76.6 %	75.3 %

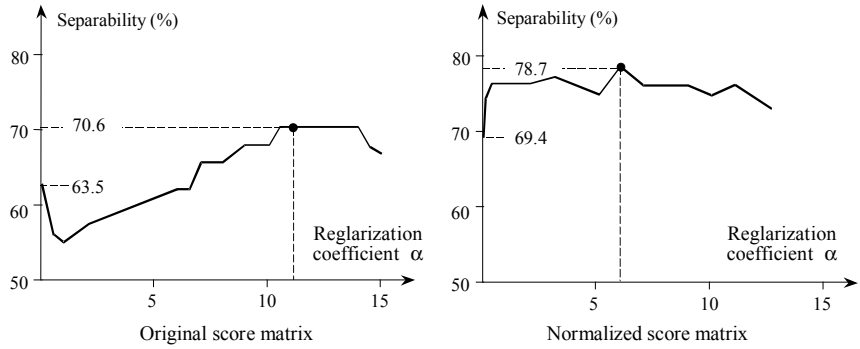


Fig. 4. Dependence of the averaged pair-wise separability over 21 fold class pairs on the regularization coefficient.

8 Conclusions

Within the bounds of the featureless approach to pattern recognition, the main idea of this work is treating the pair-wise similarity measure of objects of recognition as inner product in an imaginary Hilbert space, into which really existing objects may be mentally embedded as a subset of isolated points. Two ways of regularization of the training process follow from this idea, which contribute to overcoming the small size of the training sample. In the practical problem of protein fold class recognition, to embed the discrete set of known proteins into a continuous Hilbert space, we propose to consider as inner product the pair-wise alignment score of amino acid chains, which is commonly adopted in bioinformatics as their biochemically justified similarity measure.

9 References

1. Cortes, C., Vapnik, V.: Support-vector networks. Machine Learning, Vol. 20, No.3, 1995.
2. Vapnik, V. Statistical Learning Theory. John-Wiley & Sons, Inc. 1998.
3. Duin, R.P.W, De Ridder, D., Tax, D.M.J. Featureless classification. Proceedings of the Workshop on Statistical Pattern Recognition, Prague, June 1997.
4. Duin, R.P.W, De Ridder, D., Tax, D.M.J. Experiments with a featureless approach to pattern recognition. Pattern Recognition Letters, vol. 18, no. 11-13, 1997, pp. 1159-1166.
5. Duin, R.P.W, Pekalska, E., De Ridder, D. Relational discriminant analysis. Pattern Recognition Letters, Vol. 20, 1999, No. 11-13, pp. 1175-1181.
6. Fetrow J.S., Bryant S.H. New programs for protein tertiary structure prediction. Biotechnology, Vol. 11, April 1993, pp. 479-484.
7. Dubchak, I., Muchnik, I., Mayor, C., Dralyuk, I., Kim, S.-H. Recognition of a protein fold in the context of the SCOP classification. Proteins: Structure, Function, and Genetics, 1999, 35, 401-407.
8. Mottl, V., Dvoenko, S., Seredin, O., Kulikowski, C., Muchnik, I. Alignment Scores in a Regularized Support Vector Classification Method for Fold Recognition of Remote Protein Families. DIMACS Technical Report 2001-01, January 2001. Center for Discrete Mathematics and Theoretical Computer Science. Rutgers University, the State University of New Jersey, 33 p.
9. Durbin, R., Eddy, S., Krogh, A., Mitchison, G. Biological Sequence Analysis. Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, 1988.
10. Pearson, W. R., Lipman, D. J. Improved tools for biological sequence analysis. PNAS, 1988, 85, 2444- 2448.
11. Pearson, W. R. Rapid and sensitive sequence comparison with FASTP and FASTA. Methods in Enzymology, 1990, 183, 63-98.

Adaptive Query Shifting for Content-Based Image Retrieval

Giorgio Giacinto, Fabio Roli, and Giorgio Fumera

Department of Electrical and Electronic Engineering - University of Cagliari
Piazza D'Armi 09123 Cagliari, Italy
Tel: +39 070 675 5862 - Fax: +39 070 675 5900
{giacinto,roli,fumera}@diee.unica.it

Abstract. Despite the efforts to reduce the semantic gap between user perception of similarity and feature-based representation of images, user interaction is essential to improve retrieval performances in content based image retrieval. To this end a number of relevance feedback mechanisms are currently adopted to refine image queries. They are aimed either to locally modify the feature space or to shift the query point towards more promising regions of the feature space. In this paper we discuss the extent to which query shifting may provide better performances than feature weighting. A novel query shifting mechanism is then proposed to improve retrieval performances beyond those provided by other relevance feedback mechanisms. In addition, we will show that retrieval performances may be less sensitive to the choice of a particular similarity metric when relevance feedback is performed.

1. Introduction

The availability of large image and video archives in many applications (art galleries, picture and photograph archives, medical and geographic databases, etc.) demands for advanced query mechanisms that address the perceptual aspects of visual information, usually not exploited by traditional textual attributes search. To this end researchers developed a number of image retrieval techniques based on image content, where the visual content of images is captured by extracting features from images such as color, texture, shape, etc. [1,10]. Content based queries are often expressed by visual examples in order to retrieve from the database all the images that are *similar* to the examples. It is easy to see that the effectiveness of a content-based image retrieval (CBIR) system depends on the choice of the set of visual features and on the choice of the similarity metric that models user perception of similarity. In order to reduce the gap between perceived similarity and the one implemented in content-based retrieval systems, a large effort in research has been carried out in different fields, such as pattern recognition, computer vision, psychological modeling of user behavior, etc. [1,10].

An important role in CBIR is played by user interactivity. Even if features and similarity metric are highly suited for the task at hand, the set of retrieved images may partially satisfy the user. This can be easily seen for a given query if we let different

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 337–346, 2001.

users mark each of the retrieved images as being "relevant" or "non-relevant" to the given query. Typically different subsets of images are marked as "relevant", the intersection of subsets being usually non-empty. It is easy to see that the subset of "relevant" images as well as those marked as "non relevant" provide a more extensive representation of user needs than the one provided by the original query. Therefore this information can be fed back to the system to improve retrieval performances. A number of techniques aimed at exploiting the *relevance* information have been proposed in the literature [4-8,11]. Some of them are inspired from their counterpart in text retrieval system, where relevance feedback mechanisms have been developed several years ago, while other techniques are more tailored to the image retrieval domain. Another distinction can be made between techniques aimed at exploiting only the information contained in the set of "relevant" images and techniques aimed at exploiting information contained both in "relevant" and "non-relevant" images.

Two main strategies for relevance feedback have been proposed in the literature of CBIR: query shifting and feature relevance weighting [4-8]. Query shifting aims at moving the query towards the region of the features space containing the set of "relevant" images and away from the region of the set of "non-relevant" images [4-5,7]. Feature relevance weighting techniques are based on a weighted similarity metric where relevance feedback information is used to update the weights associated with each feature in order to model user's need. [4-6,8]. Some systems incorporated both techniques [4-5].

In this paper an adaptive technique based on a query shifting paradigm is proposed. The rationale behind the choice of the query shifting paradigm is that in many real cases few "relevant" images are retrieved by the original query because it is not "centered" with respect to the region containing the set of images that the user whishes to retrieve. On the other hand, feature weighting mechanisms implicitly assume that the original query is in the "relevant" region of the feature space, so that a larger number of relevant images can be retrieved modifying the similarity metric in the direction of the most relevant features. In our opinion feature weighting can be viewed as a complementary technique to query shifting. This opinion is currently shared by many researchers [4-5].

In section 2 a brief overview on relevance feedback techniques for CBIR is presented and the rationale behind our choice of the query shifting mechanism is discussed. The proposed relevance feedback mechanism is described in section 3. Experiments with two image datasets are reported in section 4 and results show that retrieval performances may be less sensitive to the choice of a particular similarity metric when relevance feedback is performed.. Conclusions are drawn in Section 5.

2. Relevance Feedback for CBIR

Information retrieval system performances are usually improved by user interaction mechanisms. This aspect has been thoroughly studied for text retrieval systems some decades ago [9]. The common interaction mechanism is relevance feedback, where documents marked as being "relevant" are fed back to the system to perform a new search in the database. However techniques developed for text retrieval systems need

to be suitably adapted to content based image retrieval, due to differences both in number and meaning of features and differences in similarity measures [7,11]. Usually in text retrieval systems each possible term is treated as a feature, and search is performed by looking for documents containing similar terms. On the opposite CBIR systems are usually designed with a small set of features suited for the image domain at hand. Similarity between documents thus is measured in terms of the number of "matching" terms. If D and Q represent the feature vectors related to two documents, similarity can be measured with the *cosine* metric

$$\mathrm{SIM}(D,Q) = \frac{D \cdot Q}{\|D\| \, \|Q\|} \tag{1}$$

If documents retrieved using query Q are marked as being "relevant" and " non-relevant" to the user, then a relevance feedback step can then be performed using the standard Rocchio formula [9]:

$$Q_1 = \alpha Q_0 + \beta \left(\frac{1}{n_R} \sum_{D_m \in D_R} \frac{D_m}{|D_m|} \right) - \gamma \left(\frac{1}{n_{N-R}} \sum_{D_m \in D_{N-R}} \frac{D_m}{|D_m|} \right) \tag{2}$$

where Q_0 is the query issued by the user, subscript R is for "relevant documents" while subscript N-R is for "non-relevant" documents. The new query Q_1 is obtained by a linear combination of the "mean" vectors of relevant and non relevant documents, so that Q_1 is close to the mean of relevant document and far away from the non-relevant mean. The three parameters α, β and γ are usually chosen by experiments. As an example, standard Rocchio has been implemented in [7] with a suitable normalization of image features designed to measure similarity by the cosine metric.

Another approach to relevance feedback is based on relevance feature weighting [5-6,8]. This scheme assume that the query is already placed in the "relevant" region of the feature space and that the cluster of relevant images is stretched along some directions of the feature space. Relevance feedback is thus called to improve performances by stretching the neighborhood of the query in order to capture a larger number of relevant images. A number of different weighting schemes have been proposed in the literature tailored to different similarity metrics. Moreover different techniques for estimating feature relevance from the set of relevant images have been also proposed [5-6,8].

In our opinion the two paradigms for relevance feedback, namely query shifting and feature relevance weighting, have complementary advantages and drawbacks. Query shifting mechanisms seems to be more useful when the first retrieval contains few relevant images. In this case the user may have queried the database using an image sample located near the "boundary" of the "relevant" region in the feature space. More relevant images can then be retrieved by moving the query away from the region of non-relevant images towards the region of relevant images. In this case feature relevance estimation may be too inaccurate for the lack of enough relevant images and the neighborhood may be stretched in a way that does not reflect the actual shape of the relevant region in the feature space. On the other hand when the query is able to capture a significant number of relevant images, then it should be

more effective to refine the query by some feature weighting mechanism than moving the query away. Some papers provided solutions that combine both methods [4-5]. Further discussion on the combination of the two paradigms is beyond the scope of this paper.

3. An Adaptive Query Shifting Mechanism

3.1 Problem Formulation

As a consequence of the above discussion on the main relevance feedback paradigms used in CBIR, it can be pointed out that the effectiveness of each relevance feedback technique relies upon some hypotheses on the distribution of "relevant" images in the feature space. In this section we outline the hypotheses upon which our relevance feedback mechanism is based.

Let us assume that the image database at hand is made up of images whose content exhibit little variability. This is the case of specific databases related to professional tasks. This kind of databases are often referred to as "narrow domain" databases as opposed to "broad domain" image databases, where the content of images in the database exhibit an unlimited and unpredictable variability [10]. It is easy to see that the gap between features and semantic content of images can be made smaller for narrow domain databases than for broad domain databases. Therefore a couple of images that the users judges as being similar each other, are often represented by two near points in the features space of narrow domain databases.

Let us also assume that the user aims at retrieving images belonging to a specific class of images, i.e performs a so-called "category search" [10]. Category search is often highly interactive because the user may be interested to refine the initial query to find the most suitable images among those belonging to the same class.

Finally, even if many studies in psychology have discussed the limits of the Euclidean distance as a similarity measure in the feature space, nevertheless the Euclidean model has several advantages that make it the most widely employed model [1]. Therefore we will assume that Euclidean distance is used to measure similarity between images in the feature space. It is worth noting that most of the current systems have relied upon this querying method [1].

It is quite clear from the above that the retrieval problem at hand can be formulated as a *k-nn* search in the feature space. Let I be a feature vector representing an image in a d–dimensional feature space. Let Q be the feature vector associated with the sample image used to query the database. The retrieval system then retrieves the k nearest neighbors of Q and present them to the user. The user then mark each retrieved image as being "relevant" or "non-relevant". Let I_R and $I_{N\text{-}R}$ be the sets of relevant and non-relevant images respectively, I_R and $I_{N\text{-}R}$ belonging to the *k-nn* neighborhood of the initial query Q. This information is used in a relevance feedback step to compute a new query point in the feature space where a new *k-nn* search must be performed.

3.2 Adaptive Query Shifting

Let m_R and $m_{N\text{-}R}$ be the centroids of the feature vectors of relevant and non-relevant images respectively, belonging to the *k-nn* neighborhood of query Q:

$$m_R = \frac{1}{k_R}\sum_{I \in I_R} I, \quad m_{N-R} = \frac{1}{k_{N-R}}\sum_{I \in I_{N-R}} I \tag{3}$$

where k_R and $k_{N\text{-}R}$ are the number of relevant and non-relevant images respectively. Clearly $k_R+k_{N\text{-}R}=k$.

Let D_{max} be the maximum distance between the query and the images belonging to the neighborhood of Q, $N(Q)$, defined by the k-nn search, i.e.,

$$D_{\max} = \max_{I_n \in N(Q)} \|Q - I_n\| \tag{4}$$

We propose to compute the new query Q_{new} according to the following formula

$$Q_{new} = m_R + \left(\frac{1}{2} + \frac{k_{N-R}}{k}\right)\frac{\alpha D_{\max}}{\|m_R - m_{N-R}\|}(m_R - m_{N-R}) \tag{5}$$

i.e. Q_{new} is on the line linking the two means m_R and $m_{N\text{-}R}$ at a distance equal to $\left(\frac{1}{2} + \frac{k_{N-R}}{k}\right)\alpha D_{\max}$ from the mean m_R.

This formulation of Q_{new} has some elements in common with the formulation of the decision hyperplane between two data classes, ω_i and ω_j, with normal distributions [2]. This hyperplane is orthogonal to the line linking the means and passes through a point x_0 defined by the following equation:

$$x_0 = \mu_i - \left(\frac{1}{2} - \frac{\sigma^2}{\|\mu_i - \mu_j\|^2}\ln\frac{P(\omega_j)}{P(\omega_i)}\right)(\mu_i - \mu_j) \tag{6}$$

under the assumption of statistically independent features with the same variance σ^2. When the prior probabilities are equal, x_0 is halfway between the means, while it moves away from the more likely mean in the case of different priors. In x_0 the posterior probabilities for the two classes are equal, while points with higher values of posterior probability for class ω_i are found by moving away from x_0 in the $(\mu_i - \mu_j)$ direction (if we move in the opposite direction, higher posteriors for ω_j are obtained).

From the above perspective the relevance feedback problem can be formulated as follows. The sets of relevant and non-relevant images in $N(Q)$ are the two "class" distributions with means m_R and $m_{N\text{-}R}$ respectively. The fractions of relevant (non-relevant) images in $N(Q)$ can be interpreted as the "prior" probabilities of finding relevant (non-relevant) images in the neighborhood of $N(Q)$. Even if we cannot assume normal distributions in $N(Q)$, it is reasonable to assume, in agreement with eq. (6), that the "decision surface" between the two "classes" is close to the mean of relevant images if the majority of images belonging to $N(Q)$ are non-relevant to the user. The reverse is true if the majority of images belonging to $N(Q)$ is relevant. Therefore according to eq. (5) Q_{new} is placed on the line linking the two means, on the opposite side of $m_{N\text{-}R}$ with respect to m_R, at a distance from m_R proportional to the "prior" probability of non-relevant images in $N(Q)$. From the above discussion it is easy to see that this choice of Q_{new} is aimed to keep the k-nn region of Q_{new} away from

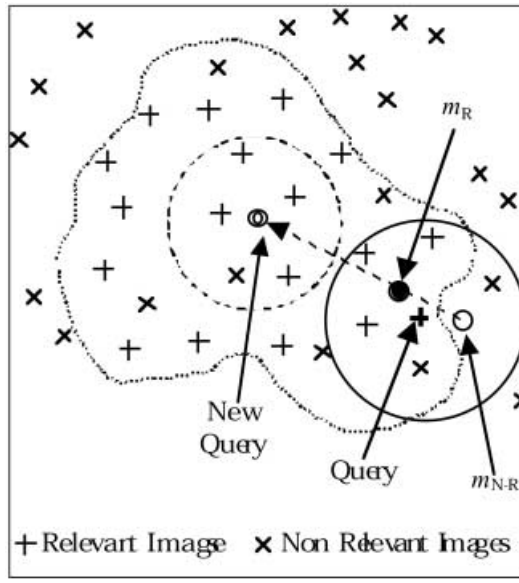


Fig. 1. A qualitative representation of the proposed query shifting method. The initial query and the related *k-nn* search (k = 5 in this example) is represented (continuous line). A new query is computed according to equation (5), and the new *5-nn* neighborhood is considered (dashed line). The boundary of the cluster of relevant images is represented by a dotted line.

the above "decision surface" and, at the same time, to put Q_{new} in a region with a high probability of finding relevant images.

It is worth noting that the aim of relevance feedback mechanisms is to explore different areas of the feature space in order to find a large number of images that are relevant to the user. To this end the selection of m_R as the new query represent a possible choice, but this choice does not take into account the information on the distribution of non-relevant images [9]. If relevant images are clustered in the feature space, then the distribution of non relevant images, summarized for example by m_{N-R}, can help in moving away from the region of non-relevant images to a region containing a large number of relevant images. Figure 1 represent qualitatively such a situation. Images relevant to the user are clustered, the cluster containing also some non-relevant images. The first k-nn search (for the sake of simplicity we have selected k = 5) retrieves 3 relevant images and 2 non relevant images. Then a new query point in the feature space is computed according to equation (5) and the related 5-nn is computed. In this second iteration all the retrieved images are relevant to the user.

Finally, it is worth explaining the term αD_{max} in eq. (5). By definition in eq. (4) D_{max} is the radius of the hypersphere containing $N(Q)$, while α is a parameter ≤ 1 chosen by experiments. This terms plays a role analogous to that of the variance σ^2 in eq. (6), i.e. it takes into account the distribution of images in $N(Q)$.

3.3 Discussion

The following considerations are aimed to point out on what extent the proposed approach is in agreement with the hypotheses in section 3.1:

it is reasonable to think that if category search is performed on a narrow domain database, images that are relevant to a specific query for a given user tend to be *clustered* in the feature space;

the above cluster of relevant images can be considered a small data class of the image database, the other images being non-relevant for the user's needs;

the image used to query the database may not be "centered" with respect to the above cluster;

the proposed relevance feedback mechanism exploits both relevant and non-relevant images to compute a new query "centered" with respect to the cluster of relevant images.

4. Experimental Results

In order to test the propose method and make comparisons with other methods proposed in the literature, two databases containing images from the real world have been used: the MIT database and one database from the UCI repository.

The MIT database is distributed by the MIT Media Lab at *ftp://whitechapel.media.mit.edu/pub/VisTex*. This data set contains 40 texture images that have been processed according to [7]. Images have been manually classified into 15 classes. Each of these images has been subdivided into 16 nonoverlapping images, thus obtaining a data set with 640 images. A set of 16 features have been extracted from the images using 16 Gabor filters [6] so that images have been represented in the database by a 16-dimensional feature vector.

The database extracted from the UCI repository (http://www.cs.uci.edu/mlearn/MLRepository.html) consists of 2310 outdoor images. Images are subdivided into seven data classes, e.g brickface, sky, foliage, cement, window, path and grass. Each image is represented by a 19-dimensional feature related to color or spatial characteristics.

For both dataset, a normalization procedure has been performed so that each feature is in the range between 0 and 1. This normalization procedure is necessary to use the Euclidean distance metric.

Since each database consists of a number of images subdivided in classes, reported experiments can be considered an example of category search performed on narrow domain databases and therefore are suited to test the proposed relevance feedback mechanism. In particular, for both problems, each image in the database is selected as query and top 20 nearest neighbors are returned. Relevance feedback is thus performed by marking as "relevant" those images belonging to the same class of the query and by marking as "non-relevant" all other images among the top 20. Such experimental set up let us make an objective comparison among different methods and is currently performed by many researchers [6-7].

Tables 1 and 2 report the results of the proposed method on the two selected datasets in terms of average percentage retrieval precision and Average Performance Improvement (API). Precision is measured as the ratio between the number of relevant retrievals and the number of total retrievals averaged over all queries. API is computed averaging over all queries the ratio

$$\frac{relevant\ retrievals(n+1) - relevant\ retrievals(n)}{relevant\ retrievals(n)} \tag{7}$$

where $n = 0, 1, \ldots$ is the number of feedbacks performed.

For the sake of comparison, retrieval performances obtained with other methods, namely RFM and PFRL, are also reported. PFRL is a probabilistic feature relevance feedback method aimed at weighting each feature according to information extracted from relevant images [6]. This method use the Euclidean metric to measure similarity between images. RFM is an implementation of the standard Rocchio formula for CBIR [7]. It is worth noting that in this case similarity between images has been measured with the cosine metric and consequently a different normalization procedure has been performed on the data sets in order to adapt features to the cosine metric.

The first column in tables 1 and 2 reports the retrieval performance without any feedback step. It is worth noting that differences in performances depends on different similarity metrics employed: the Euclidean metric for PFRL and the proposed adaptive query shifting method; the cosine metric for RFM. Clearly such differences affect the performances of the feedback step since different relevance information is available. This results also point out that the cosine metric is more suited than the Euclidean metric for the MIT data set, while the reverse is true for the UCI data set. Therefore if no relevance feedback mechanism is performed, retrieval performances are highly sensitive to the selected similarity metric.

The second column reports the precision of retrieval performance after relevance feedback. Regarding the proposed query shifting method, we selected values of the α parameter (see eq. 5) equal to 5/6 and 2/3 for the MIT and UCI data sets respectively. These values allowed to achieve maximum performances in a number of experiments with different values of α. The proposed method always outperforms PFRL and RFM in both data set. However it is worth noting that in the first retrieval, different sets of top 20 nearest neighbors are retrieved. Therefore each method received different relevance information, and the retrieval performances after relevance feedback reported in the second column are biased. Nevertheless it is worth making further comments on the results on the MIT data set. If no relevance feedback is performed, the cosine metric provides better retrieval performances than the Euclidean metric. On the other hand the proposed query shifting method based on the Euclidean metric was able not only to outperform the precision of the first retrieval using the cosine metric, but also to provide better performances than those obtained with RFM, which exploits a larger number of relevant images available from the first retrieval. Therefore it can be concluded that retrieval performances provided by the proposed relevance feedback method are less sensitive to the choice of the Euclidean similarity metric.

The comparison between PFRL and the proposed query shifting method points out that query shifting is more suited for relevance feedback than feature weighting when category search is performed in narrow domain databases. This result is also

confirmed by results reported in [4] where PFRL performances are improved by combining it with a query shifting mechanism. This combined method allowed to achieve retrieval performances equal to 89% and 95.5% on the MIT and UCI datasets respectively. However, it should be noted that our query shifting mechanism provides better results than the above combined method, thus confirming that a suitable query shifting mechanism is also able to outperform more complex methods.

The above conclusion is also confirmed if the average performance improvements (API) are compared, designed to measure the relative improvements with respect to performances of the first retrieval. Our method provides the largest performance improvement on both data set. In particular the advantages of the proposed method are more evident on the MIT data set.

Table 1. Retrieval Precision on the MIT data set

Relevance feedback Mechanism	1st retrieval	2nd retrieval with relevance feedback	API
RFM	83.74%	90.23%	13.53
PFRL	79.24%	85.48%	12.70
Adaptive query shifting	79.24%	**91.85%**	**33.79**

Table 2. Retrieval Precision on the UCI data set

Relevance feedback mechanism	1st retrieval	2nd retrieval with relevance feedback	API
RFM	86.39%	91.95%	15.33
PFRL	90.21%	94.56%	7.66
Adaptive query shifting	90.21%	**96.35%**	**15.68**

5. Conclusions

Relevance feedback mechanisms are essential to modern content based image retrieval because they are aimed to fill the semantic gap between user perception of similarity and database similarity metrics. Different relevance feedback methods have been proposed in the literature based on two main paradigms: query shifting and feature weighting. We discussed the advantages and disadvantages of both paradigms and concluded that query shifting is more suited for category search in narrow domain databases. We thus presented a novel query shifting mechanism and showed the hypotheses under which such an approach can improve retrieval performances. Experimental results on two image datasets showed that the proposed method is an effective mechanism to exploit relevance feedback information. In addition, reported results also pointed out that significant improvements in retrieval performances can be obtained by relevance feedback mechanisms rather than by selecting different similarity metrics.

References

1. Del Bimbo A.: Visual Information Retrieval. Morgan Kaufmann Pub (1999)
2. Duda R.O., Hart P.E. and Stork D.G.: Pattern Classification. J. Wiley & Sons (2000)
3. Faloutsos C. and Lin K.: *FastMap*: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. Proc. of the 1995 ACM SIGMOD intern.l conf. on Management of data, San Jose, CA USA (1995) 163-174
4. Hesterkamp D.R., Peng J. and Dai H.K.: Feature relevance learning with query shifting for content-based image retrieval. Proc. of the 15th IEEE Intern.l Conf. on Pattern Recognition (ICPR 2000) (2000) 250-253
5. Ishikawa Y., Subramanys R. and Faloutsos C.: MindReader: Querying databases through multiple examples. Proc. of the 24th VLDB Conf., New York (1998) 433-438
6. Peng J., Bhanu B. and Qing S.: Probabilistic feature relevance learning for content-based image retrieval. Computer Vision and Image Understanding. **75** (1999) 150-164
7. Rui Y., Huang T.S., and Mehrotra S.: Content-based image retrieval with relevance feedback: in MARS. Proc. IEEE intern.l conf. on Image Processing, Santa Barbara, CA (1997) 815-818
8. Rui Y., Huang T.S., Ortega M. and Mehrotra S.: Relevance Feedback: a power tool for interactive content-based image retrieval. IEEE Trans. on Circuits and Systems for Video Technology **8** (1998) 644-655
9. Salton G. and McGill M.J.: Introduction to modern information retrieval. McGraw-Hill, New York (1988)
10. Smeulders A.W.M., Worring M., Santini S., Gupta A. and Jain R.: Content-based image retrieval at the end of the early years. IEEE Trans. on Pattern Analysis and Machine Intelligence **22** (2000) 1349-1380
11. McG. Squire D., Müller W., Müller H. and Pun T.: Content-based query of image databases: inspirations from text retrieval. Pattern Recognition Letters **21** (2000) 1193-1198

Content-Based Similarity Assessment in Multi-segmented Medical Image Data Bases

George Potamias

Institute of Computer Science, Foundation for Research and Technology – Hellas (FORTH), P.O. Box 1385, GR-711 10 Heraklion, Crete, Greece
potamias@ics.forth.gr
and
Dept. of Computer Science, University of Crete, P.O. Pox 1470, GR-714 09 Heraklion, Crete, Greece

Abstract. Image database systems and image management in general are extremely important in achieving both technical and functional integration of the various clinical functional units. In the emerging '*film-less*' clinical environment it is possible to extend the capabilities of diagnostic medical image techniques and introduce intelligent content-based image retrieval operations, towards 'evidence-based' clinical decision support. In this paper we presented an integrated methodology for content-based retrieval of multi-segmented medical images. The system relies on the tight integration of clustering and pattern- (similarity) matching techniques and operations. Evaluation of the approach on a set of indicative medical images shows the reliability of our approach.

1 Introduction

Image database systems and image management in general are extremely important in achieving both technical and functional integration of Hospital Information Systems (HIS), Radiological Information Systems (RIS), PACS, and Telemedicine Systems due to the technical constraints imposed by the volume and information density of image data.

Current research on visual information systems and multimedia databases raises a number of important issues, including the need for *query* methods, which support *retrieval* of images by content [7], [11]. At the same time, the rapid growth of popularity enjoyed by the World Wide Web during the last years, due to its visual nature and information retrieval capabilities, resulted in the development of systems that provide network-transparent information services based on pictorial content [12], [13]. In this vast, dynamic information infrastructure, the development of medical information systems with advanced browsing and navigation capabilities and a visual query language supporting content-based similarity queries will play an increasingly important role in medical training, research, and clinical decision making.

P. Perner (Ed.): MLDM 2001, LNAI 2123, pp. 347–361, 2001.

Content-based image retrieval is not only a complex task, but also difficult to define. Pictures are '*beyond words*', and as it is stated in a recent review study about content-based image retrieval in [14], "… *Pictures have to be seen and searched as pictures: by object, by style, by purpose*". But, *how* the computer may see an image, in other words, what are the essential information items that should be extracted? How they are to be extracted, and finally, *how an image is to be interpreted?* Towards these objectives various approaches has been proposed. They range from search and browsing *by association*- in order to find interesting things [15], and *target search*- in order to match a pre-specified image description [5], to *category search* which aims at retrieving an arbitrary *image representative* of a specific *class* [17], [18]. In category search, the user have available a group of images and the search is for additional images of the same class. The key-concept in category search is the definition of *similarity*.

This paper presents an approach to *content-based retrieval and similarity assessment of multi-segmented medical images*. Our approach is based on the tight integration of *clustering* and *pattern-matching* techniques following three steps.

- *Segmentation.* The images are segmented and sets of spatial, geometric/shape and texture characteristics are extracted.
- *Clustering.* The resulted segments are clustered using a Bayesian clustering system. This step aims towards the identification of similar-groups of 'Regions Of Interest' (ROIs) in medical images. With this operation we are able to identify a representative segment for each of the formed clusters (and for each of the images as well), and in a way to assess a more natural interpretation of the images in the database (e.g., "this group of images are of tumor-X-class").
- Classification. A pattern-matching operation is activated in order to compute the similarity of query images with the representative segments of the stored images. The result is the classification of query images to one of the formed clusters, and the identification of the most-similar images in the database.

The paper is organized as follows. Next section refers to the segmentation and image representation issues. In Section 3 the segments' clustering operation is presented. Section 4 presents the details of the images' pattern- (similarity) matching and classification operations. Section 5 presents a series of experiments on an indicative database of CT tumor-brain images. In the last section, we conclude and propose dimensions for further research and work.

2 Segmentation and Representation of Images

Segmentation. The first step in the content-based similarity assessment of medical images is the *segmentation*, i.e., the *partitioning* of medical images. Partitioning of the image aim at obtaining more *selective features*. The segmentation of images is performed within the I^2C system. I^2C (*Image to Content*) is an image management system, which has been developed by the Computer Vision and Robotics group of the Institute of Computer Science – FORTH [10], [11]. The I^2C environment offers *Regions Of Interest*- ROI identification services. It captures their content by applying

one of the available segmentation algorithms and gets their *feature-based* descriptions. The generated image content descriptors include *spatial*, *geometric/shape* and *texture* features. In the course of the current study we focus on the set of features summarized in table 1, below. The extracted feature-based descriptions compose a set of '*logical images*'.

Table 1. Spatial, Geometric/Shape and Texture features used

Feature	# Values / Type	Meaning
Spatial		
Center (X,Y)	2 /real	X, Y coordinates of segment's center
Upper_max (X,Y)	-‖-	X, Y coordinates of upper-most-left segment's point
Lower_min (X,Y)	-‖-	X, Y coordinates of lower-most-right segment's point
Geometric/Shape		
Area	1 / real	The area occupied by the segment
Compactness	-‖-	The compactness of the segment
Orientation	-‖-	A number indicating the orientation of the segment
Roundness	-‖-	A number indicating the roundness of the segment
Texture		
Contrast	16 / real	From *Concurrence Matrix* calculations- 4 distances in
*IDF**	-‖-	{1, 3, 5, 7}, and 4 angles in {0, 45, 90, 135}. A total
Entropy	-‖-	of 16 numbers are computed for each feature (= 4 dists
Correlation	-‖-	4 angles) [6], [9]

**IDF: Inverse Difference Moment*

Representation. When a diagnostician faces a medical image associated with a specific patient tries to identify basic characteristics of the image that may guide him/her to a confident diagnosis or, follow-up actions. Doing so, they recall- from their accumulated expertise, other indicative images that seem to be *relevant* to the case at hand. This mental process presupposes the existence of some '*model*' images each of which is more-or-less associated with a specific pathology. The identification and formation of such model images is based on some form of expert background knowledge able to define and cope with:

- *indicative* regions of medical images associated with potential pathologies, as identified by confirmed clinical symptoms and findings, and
- *descriptive* features of images, i.e., the features that mostly characterize a specified clinical-state.

Trying to approach and simulated the human's mental process, and equipped with a repository of segmented medical images, the first step is to identify and form models of *similar regions* in the images. Such groups of regions could be indicative of potential pathological states. Confronted with a repository of multi-*segmented* images the problem of how to compute similarities between images of varying number of segments is crucial. Even with the aid of an I^2C-like system- with ROI identification capabilities, the following problems raise:

- which segments are the most *representative* for an image or, for a particular pathology?
- which features are the most *descriptive* of the content of an image?

The discussion above forced us to represent each image as a set of segments. That is, from the space of images we are moving to a more abstract space, the space of images' segments. So, each image is represented and encoded as a series of segments:

$$I_i = <s_{i;1}, s_{i;2}, \ldots s_{i;k}>,$$

where $s_{i;j}$ is the j^{th} segment of image i. The number of segments k, may vary for each of the different images in the store.

Now, from the segmentation process each segment is described by a set of features (like the ones referred shown in table 1). So, an ordered vector of feature-values is used to represent each segment:

$$s_j = <f_{j;1}, f_{j;2}, \ldots f_{j;m}>,$$

where, $f_{j;l}$ stands for the value of feature l in segment j. The number of features m is fixed for all segments. In the sequel we will address issues related to the ignorance and restriction of the feature-space, i.e., the problem of *feature-selection*.

The correspondence of images to their segments is defined by the following function:

$$g: I \rightarrow 2^S,$$

where, I the set of images and S the set of segments.

3 Clustering Segments of Images

In vast majority of applications, no structural assumptions are made, all the structure in the classifier being *learnt* from the data. This process is known as *statistical pattern recognition*. The training set is regarded as a sample from a population of possible *examples* (*observations* or, *objects*), and the statistical similarities of each class extracted, or more precisely the significant differences between potential classes are found.

Clustering represents a convenient method for organizing a large set of data so that retrieval of information may be made more efficiently. Detecting patterns of similarity and differences among objects under investigation may provide a very convenient summary of the data. The problem that clustering techniques address may be stated broadly as [2]:

> *Given a collection of 'n' objects or events each of which is described by a set of 'm' characteristics, variables or features, derive a useful division into a number of classes. Both the number of classes and the properties of the classes are to be determined.*

In our case, the set of objects consists of all images' segments and clustering aims to:

- discover *coherences* in the set of segments. Similar groups of segments are identified, and each group is linked with some potential interpretation, i.e., tumor-type. This could be considered as the *recognition* or, *abstraction* phase in image analysis and understanding where, each image is linked to one or more general classes or, types,
- capture *representative* regions in the images. The importance of the different segments is measured according to their strength and utility in describing the recognized images' classes. This could be considered as the *focus-of-attention* phase in image understanding where, for each image potential ROIs to focus-on are identified,
- identify the *descriptive* power of features used to describe the images. The features that seem more descriptive (and/or discriminant) with respect to the semantics of the query and the exploration task at hand are identified (e.g., identify regions of the same shape vs. identify regions with the same texture). This could be considered as the *interpretation* phase in image-analysis where, each image is assigned a more specific meaning according to a set of natural features used to describe it.

For the clustering operation we rely on one of the most known and reliable clustering system from the machine learning research namely, the Autoclass system [4]. *Autoclass* is an unsupervised classification system based on Bayesian theory. Rather than just partitioning cases, as most clustering techniques do, the Bayesian approach searches in a model-space for the "*best*" class descriptions. A best classification optimally trades off predictive accuracy against the complexity of the classes, and so does not "*overfit*" the data. Autoclass does not rely on specific similarity metrics. Instead, it seeks for the *most probable* set of class descriptions given the data and *prior* expectations. Furthemore, Autoclass does not require a pre-specified number of clusters to form. For more details on the specifics of the Autoclass system the reader may refer to [4], [8]. The reasons for using Autoclass as the utility-clustering system are:

- the number of clusters to be discovered is not pre-specified. In our case, the active set of images give us no indication about the potential types or, classes (e.g., pathologies) present, and
- it offers some useful measures for evaluating: (a) the *strength of clustering*, i.e., how well the segments of images are grouped, (b) the *strength of each cluster*, i.e., how much similar are the segments present in each cluster, (c)the *influence* of each feature for the overall clustering, i.e., the power and the importance of each of the used features for the resulted grouping, and (d) the *influence* of each feature for each of the discovered/ formed clusters, i.e., the power that each feature exhibits in order to represent a cluster and so, the importance of them for describing the clusters.

4 Similarity Assessment of Multi-segmented Medical Images

From the segmentation and clustering operations we have at our disposition a set of segments' clusters. The clusters indicate some form of coherences present in the active set of images. The next task is to design a procedure able to *classify* a test or, *query image* to one or more of the formed classes.

Towards this end we rely on *instance-based* (or, *lazy*) *learning* operations like the ones introduced in [1]. The basic notion in instance-based learning and classification approaches is the *distance* between instances. Although there have been many distance functions proposed, by far the most commonly used is the Euclidean distance function. One weakness of the basic Euclidean distance function is that if one of the input attributes has a relatively large range, then it can overpower the other attributes. For example, in a domain-application with features, f_i and f_j where, f_i can have values from 1 to 1000, and f_j has values only from 1 to 10, then f_j's influence on the distance function will usually be overpowered by f_i's influence. Therefore, distances are often *normalized.*

Distance between feature-values. Given a feature, f_i with values $f_{i;a}$ and $f_{i;b}$, their distance is computed by formula 1, below.

$$d(f_{i;a}, f_{i;b}) = \frac{|f_{i;a} - f_{i;b}|}{range\ (f_i)}, \tag{1}$$

where, $range(f_i) = max(f_i)\text{-}min(f_i)$, and $max(f_i)$, $min(f_i)$ are the maximum and minimum values of feature f_i, respectively.

Formula 1 is a *normalization* on the difference between the actual feature-values. The normalization serves to scale the attribute down to the point where differences are almost always less than one. By dividing the distance for each feature by the *range* of that feature, the distance for each feature is in the approximate range 0..1. So, features with relatively high values (e.g., area of a segment), and features with relatively low values (e.g., entropy of a segment) are uniformly treated (i.e., high valued features does not overpower lower valued feature). In order to avoid outliers, it is also common to divide by the standard deviation instead of range. We do not follow the standard normal-distribution normalization because there is no in-advance evidence that feature-value samples follow the normal distribution. Furthermore, studies on normalized value-difference metrics, like the one introduced by formula 1, have shown the reliability and efficiency of the approach [19].

Distance between segments. Given two segments, s_a, and s_b, their distance is computed by formula 2, below.

$$d(s_a, s_b) = \frac{\sum_{i=1}^{m} d(f_{i;a}, f_{i;b})}{m}, \tag{2}$$

where, m is the total number of features used to represent the segments.
In its kernel, formula 2 is a Euclidean formula. It resembles the Manhattan/ city-block distance formula (i.e., $\sum_{i=1}^{m} |f_{i;a} - f_{i;b}|$), and as it is noted in [3], the Euclidean and

Manhatan/ city-bloack metrics are equivalent to the Minkowskian *r*-distance metric (i.e., $[\sum_{i=1}^{m} |f_{i;a} - f_{i;b}|^{r}]^{1/r}$) with r=2 and 1, respectively.

Mean cluster's segment. Given a cluster of segments c, the mean segment of the cluster is an ordered vector composed by the mean-values of all of its feature values,

$$\mu(f_c) = <\mu(f_{c;1}), \mu(f_{c;2}), \ldots \mu(f_{c;m})>$$

The mean-value of feature i for cluster c, is computed by formula 3, below.

$$\mu(f_{c;i}) = \frac{\sum_{i=1}^{|c|} f_{i;a}}{|c|}, \tag{3}$$

where, $|c|$ is the total number of segments in cluster c.

Representative segments and images. Given a cluster c, the most representative segments of the cluster MRS_c, are the ones that exhibit the *minimum distance* from its mean segment. That is,

$$MRSc = \arg\min_{a \in c} \{d(s_a, \mu(f_c))\}, \tag{4}$$

The images to which the segments in MRS_c belong are considered as the *most representative images* of the cluster.

4.1 Query Image Classification

Having identified a set of clusters, accompanied with their most representative segments and images, the task of assessing the *similarity* of an a query image with a collection of images could be accomplished.

Assume a query image I_q. Furthermore, assume that I_q is passed from a segmentation operation resulting into a set of $|I_q|$ segments. The distance of I_q with cluster c is computed by formula 5, below.

$$dist(I_q, c) == \frac{\sum_{a \in Iq} \sum_{b \in MRSc} d(s_a, s_b)}{I_q \times |MRS_c|}, \tag{5}$$

where, s_a stands for the segments in the query image I_q; s_b for the most representative segments of cluster c; and $|MRS_c|$ for the number of most representative segments in the cluster c.

We may augment the above formula with some form of background knowledge reflecting the *preference* imposed on some segments (e.g., *focus on specific pathological parts of images*),

$$dist(I_q, c) == \frac{\sum_{a \in Iq} \sum_{b \in MRSc} w(s_a) \times d(s_a, s_b)}{I_q \times |MRS_c|}, \tag{6}$$

where, $w(s_a)$ is the *weight* assigned to query-segment s_a. The *weighted distance formula*, offer image-analysts the ability to *focus* their *attention* on specific ROIs of an image and by that, acquire similar images with respect to their personal *preferences*.

After computing the distance of the query image with all clusters, the image is considered to be more similar the most representative image(s) of the cluster with which it exhibits the minimum distance. The final outcome is an ordering of the images in the active database according to their similarity to the query image. Providing that a natural interpretation (i.e., class/type; for example *'tumor-type-X'*) is assigned to the cluster then, the query image may be assigned the same interpretation.

The overall architecture of the presented content-based similarity assessment of multi-segmented images is shown in figure 1, below.

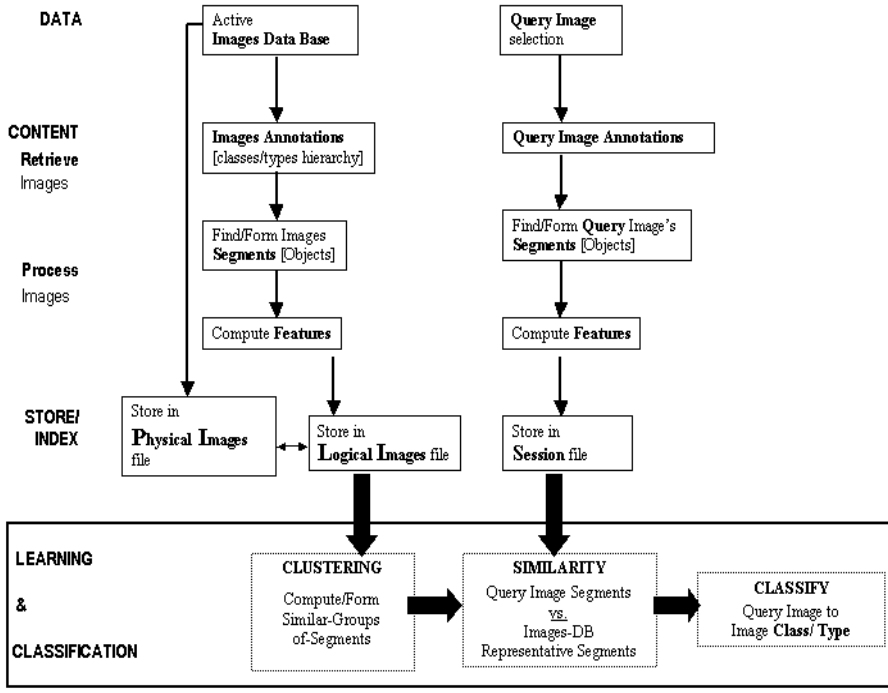


Fig. 1. Algorithmic components and operations' flow in an content-based image retrieval and querying system

5 Experimental Results

In order to test our approach to content-based similarity assessment of multi-segmented medical images, we performed a series of experiments on an indicative dataset of eleven- (11) CT brain-tumor (bt) images. The images were segmented with the aid of the I^2C system, resulting in a varying number of segments for each of the images (from 2 to 4). The features used to describe each of the images are the ones referred in table 1, section 2.

The rational behind the use of this (relatively small) set of medical images for the evaluation of our approach follows.

- The lack of a comprehensive and publicly available collection of images, sorted by class and retrieval purposes, together with a protocol to standardize experimental practices. This fact is also confirmed in a recent excellent review on content-based image retrieval [14] where, the initiation for a program for such a repository is raised and supported.
- The fact that the bt images are classified to specific tumor-types, coupled with their small number offers the ability to easily visualize the results, compare them and induce natural and comprehensive conclusions. The present case study, and the related evaluation experiments focus on the *reliability* aspects of the approach. In future work (look at the last section) we plan to test and evaluate our approach on a much larger set of images carefully located and collected. This will give us the ability for more realistic tests on the performance and effectiveness of our approach (i.e., scalability).

Content-based image retrieval and feature-selection. The performed experiments compose a methodology that addresses the *feature-selection* problem. Feature-selection is an active area of research in image-recognition and classification tasks with a fundamental question: "*which are the most appropriate features for the description of an image*?" [16].

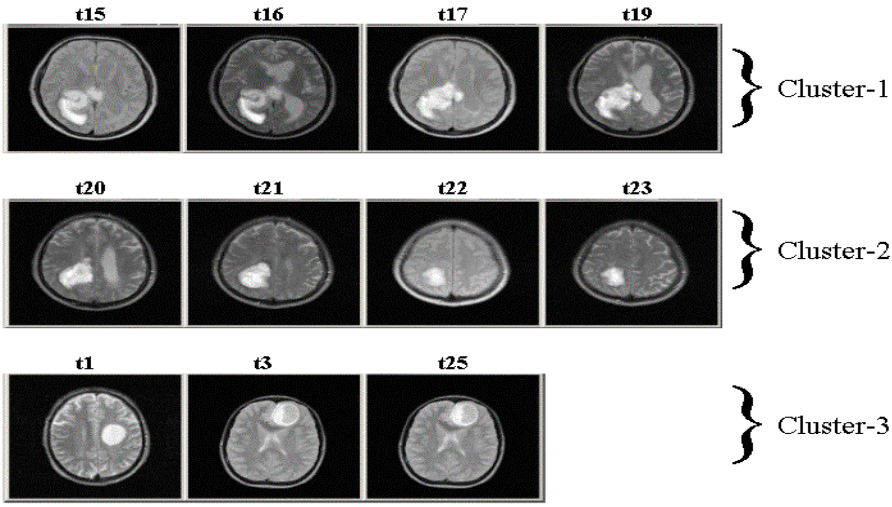
We do not claim that content-based image retrieval is the sole-key to image recognition and classification. In most of the cases content-description features (e.g., area or, entropy) are 'semantically empty'! This is because, users seek for semantic similarity, but the database of segmented-images can only provide similarity by data processing; the so-called '*semantic gap*' in content-based image retrieval [14].

Nevertheless, research and case studies on methodologies that helps to identify discriminant and descriptive image descriptors is a step towards filling this gap. Furthermore, carefully selected domain-specific images that carry pre-defined and pre-specified annotations would ease the interpretation of the data-driven similarity and classification operations. The set of bt medical images, on which the following experiments were performed, meets these specifications.

Experiment-1 [Spatial features]**.** The 11 bt images were classified relying solely on their spatial characteristics. The Autoclass clustering system offers the ability to perform a clustering operation ignoring some of the features. So, the features used for this experiment were: the *center's* and the *upper / lower* coordinates of the segments. The results are shown in figure 2, below.

A total of three- (3) clusters were identified. The *influential* power of the used spatial features with respect to the tasks of: differentiating between the clusters, and predicting the objects within a cluster are reported in table 2, below (the figures are generated by the Autoclass system; for more details on these figures the reader may refer to [Autoclass manual, http://ic.arc.nasa.gov/ ic/ projects/ bayesgroup/ autoclass/ autoclass-program.html].

(**a**) Clustering results using solely *spatial* features



(**b**) Similarity results using solely *spatial* features

Query Image	Most Similar Images	Query Image	Most Similar Images	Query Image	Most Similar Images
t15	t16 t17 t19	t20	t21 t22 t23	t1	t3 t25
t16	t15 t17 t19	t21	t20 t22 t23	t3	t25 t1
t17	t19 t15 t16	t22	t23 t20 t21	t25	t3 t11
t19	t17 t15 t16	t23	t22 t20 t21		

Fig. 2. Clustering and similarity results for experiment-1 (using only *shape* features); (a) clustering of images, (b) similarity between images (in descending order)

- *Cluster-1:* All images have a pathologic part on their low-left part. Furthermore, the similarity results indicate that the images in subgroup {t15, t16} are more similar compared to the subgroup {t17, 19} of images. This result could be confirmed from the visualized images where, the '*center'* and '*min(X,Y) coordinates*' spatial characteristics (most influential for this cluster) of the first subgroup appear closer, compared to the images of the second subgroup.
- *Cluster-2:* All images have a pathologic part on their low-left part. The images in this cluster are separated from the ones in Cluster-1 because- as it can be confirmed from the visualized images, their spatial characteristics differ at least for the '*max (X,Y) coordinates*' feature (one of the most influential for this cluster). As for Cluster-1, the similarity results indicate that images in sub-cluster {t20, t21} are more similar compared to the sub-group of images {t22, 23}.

– *Cluster-3:* All images have a pathologic part on their right part; this is the main reason that the images of this cluster are separated from the ones in Clusters-1, and 2. Sub-grouped image {t1} exhibits lower similarity figures compared to the images in subgroup {t3, t25}. This could be confirmed from the visualized images and could be attributed to the profound appearance differences for their '*center'* and '*min/max X,Y coordinates*' features.

In general, this experiment confirms, from a visualization point of view, the reliability of the presented clustering and pattern matching (similarity) approach.

Table 2. Spatial Influence of spatial features; for overall clustering and for each sole cluster; figures in italics indicate the most influential features (1 to 2 for each cluster) used

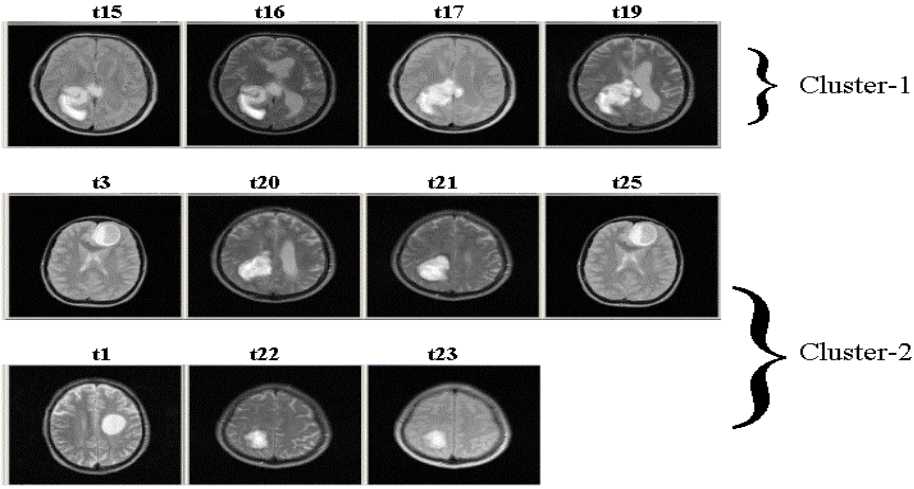
Feature	Clustering Quality	Cluster-1	Cluster-2	Cluster-3
Center	**0.83**	**2.63**	**2.91**	1.69
Min(X,Y)	**0.63**	**2.37**	1.63	1.45
Max(X,Y)	**0.74**	2.19	**2.45**	**2.48**

Experiment-2 [Geometric/Shape features]**.** The 11 bt images were classified relying solely on their shape characteristics. So, the features used for this experiment were: *area*, *compactness*, *orientation*, and *roundness*. The results are shown in figure 3, below.

A total of two- (2) clusters were identified. The *influential* power of the geometric/shape features with respect to: differentiating between the clusters, and predicting the objects within a cluster are reported in table 3, below.

– *Cluster-1:* All images are (more-or-less) of the same shape. For their pathologic regions, they share closer figures for the '*area'* and '*roundness'* features (the most influential for this cluster), compared to the respective figures for the images in cluster-2; this explains their separation from cluster-2 images. Furthermore, the similarity results indicate that the images in subgroup {t15, t16} are more similar, compared to the sub-group of images {t17, 19}. This result could be confirmed from the visualized images where, the shape of the images in the first cluster looks more similar than those in the second subgroup.
– *Cluster-2:* All images are (more-or-less) of the same shape. For their pathologic regions, they share closer figures for the '*area'* and '*compactness'* features (the most influential for this cluster), compared to the respective figures for the images in cluster-2; this explains their separation from cluster-2 images. Furthermore, the similarity results indicate that the images in subgroup {t1, t3, t25} are more similar, compared to the subgroup of images {t20, t21, t22, t23}. This result could be confirmed from the visualized images where, the shape of the images in the first subgroup looks more similar compared to those in the second subgroup, at least with reference to their '*area'* and '*roundness'* appearance.

(**a**) Clustering results using solely *geometric/shape* features



(**b**) Similarity results using solely *geometric/shape* features

Query Image	Most Similar Images	Query Image	Most Similar Images
t15	t16 t17 t19	t20	t21 t22 t23 t1 t3 t25
t16	t15 t17 t19	t21	t20 t22 t23 t1 t3
t17	t19 t15 t16	t22	t20 t21 t23 t1 t3 t25
t19	t17 t15 t16	t23	t20 t21 t22 t1 t3 t25
t1	t3 t25 t21 t22 t23 t20	t25	t20 t21 t22 t23 t1 t3
t3	t1 t25 t21 t22 t23 t20		

Fig. 3. (a) Clustering and similarity results for experiment-2 (using solely geometric/shape features); (a) clustering of images, (b) similarity between images (in descending order)

In general, this experiment confirms, from a visualization point of view, the reliability of the presented clustering and pattern matching (similarity) approach.

Table 3. Influence of geometric/shape features; for overall clustering and for each sole cluster; bold figures indicate the most influential features

Feature	Clustering Quality	Cluster-1	Cluster-2
Area	**1.00**	**1.77**	**0.65**
Roundness	**0.98**	**2.00**	0.36
Compactness	**0.92**	1.30	**0.93**
Orientation	0.47	1.08	0.06

Experiment-3 [Texture features]**.** The 11 bt images were classified relying solely on their texture characteristics. So, the features used for this experiment were: *contrast*, *IDF*, *entropy*, and *correlation*. A total of three- (3) clusters were identified. *Cluster-1* = {t3, t20, t25}, *Cluster-2* = {t1, t22, t23}, and *Cluster-3* = {t15, t16, t17, t19, t21} (look at the figures above). We do not proceed into visualizing the clustering results because, on the absence of some pre-existing knowledge for the real diagnostic-classification of the images (i.e., some form of natural interpretation for the images' texture characteristics), it would be really difficult to interpret them. The basic conclusion from this experiment concerns the most influential features. The '*entropy'* feature was the one that gets the highest influential values.

Table 4. 'Leave-one-out' assessment results on content-based similarity of images (the most-influential features, and the weighted-similarity formula were used)

Query Image	Most-Similar Images (Upper 30%)
t1	t3 t25
t3	t3 t1
t15	t16 t17
t16	t15 t17
t17	t15 t16
t19	t17 t20
t20	t19 t17
t21	t22 t23
t22	t21 t23
t23	t21 t22
t25	t3 t1

Experiment-4 [Feature Selection: most influential features and weighted similarity]. In this experiment we used <u>just</u> the most-influential features, as indicated by the previous experiments, that is: '*center'*, '*min/max (X,Y) coordinates*', '*area'*, '*roundness'*, '*compactness'*, and '*entropy'*. So from a set of eleven features (originally used to describe the images' segments; see table 1, section 2) a set of six-(6) features is now used. The restricted set of features seems more appropriate for the (relatively) small set of available medical images.

In this experiment we activated the weighted-similarity formula, *wdist* (look at section 3.1), in order to assess the similarity of query images. Doing so, the pathologic segment of the query images were assigned <u>double</u> of the weight assigned to the other segments.

For the evaluation we followed a '*leave-one-out*' process. That is, we run our system eleven times; at each time we used ten images and used the one left as the query image. The results are summarized in table 4, above, and could be considered as quite satisfactory (as may be confirmed by the visualized images), indicating the reliability of the approach.

6 Conclusion and Future Work

In this paper we presented an integrated methodology for content-based retrieval and similarity assessment between multi-segmented medical image stores. The methodology relies on the tight integration of clustering and pattern- (similarity) matching techniques and operations. The medical images are segmented by activating appropriate automated or, semi-automated segmentation operations. The result is a repository of segments linked with the images that contain them. Then, the set of segments is passed from a clustering operation (we used the Autoclass system) in order to partition them into groups that potentially indicate underlying pathologic types and classes. Finally, with an appropriate similarity-matching process we are able to compute the similarity of query images with images in the active database. The methodology was evaluated on a set of indicative tumor-brain CT images. The results are satisfactory, indicating the reliability of our approach.

The large number of medical images currently generated by various diagnostic modalities has made their interpretation as well as their management a difficult and tedious task. In the emerging '*film-less*' clinical environment it is possible to extend the capabilities of diagnostic medical image techniques. In this context, the provision of services that support content-based access, management, and processing of medical images will enhance clinical decision-making tasks towards '*evidence-based*' medicine. The presented methodology and related pattern-matching operations aims towards this goal.

Our future research and development plans include: (a) experimentation with large (and potentially more informative and organized) image data sets, in order to refine the introduced methodology and metrics, and (b) integration of the presented methodology and system within the I^2C content-based management and retrieval system.

Acknowledgment. We thank the members of the Robotics and Computer Vision Lab at the Institute of Computer Science, FORTH, Heraklion, Crete, for putting at our disposal the I^2C medical-image retrieval system, and the set of the segmented brain-tumor images. We would also like to thank the anonymous reviewers for their remarks and comments.

References

1. Aha, D.W., Dennis, K., and Albert, M.K. (1991). Instance-Based Learning Algorithms. Machine Learning, 6, 37-66.
2. Babovic, V. (1999). Seabed recognition Using Neural Networks. D2K Tech. Report – 0399-1, chapter 3, Danish Hydraulic Institute.
3. Batchelor, B.G. (1978). Pattern Recognition: Ideas in Practice. New York: Plenum Press, pp. 71-72.
4. Cheeseman, P., and Stutz, J. (1996). Bayesian Classification (AutoClass): Theory and Results. In U. M. Fayyad, G. Piatetsky-Shapiro, P. Smith, and R. Uthurusamy (eds.), Advances in Knowledge Discovery and Data Mining, pp. 153-180, AAAI/MIT Press.

5. Cox, I., Miller, M., Minka, T., and Papathomas, T. (2000). The Bayesian Image Retrieval System, PicHunter: Theory, Implementation, and Psychophysical Experiments. IEEE Trans. Image Processing, 9(1), pp. 20-37.
6. du Buf, J., Kardan, M., and Spann, M. (1990). Texture Feature Performance for Image Segmentation. Pattern Recognition, 23, 291-309.
7. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P. (1995). Query by Image and Video Content: The QBIC System. IEEE Computer, 28(9), pp. 23-32.
8. Hanson, R, Stutz, J., and Cheeseman, P. (1991). Bayesian Classification Theory. Tech. Report FIA-90-12-7-01, NASA Ames Research Center, Artificial Intelligence Branch.
9. Haralick, M. (1979). Statistical and Structural Approaches to Texture. Proc. of the IEEE, 67(5), pp. 786-804.
10. Orphanoudakis, S., Chronaki, C., and Kostomanolakis, S. (1994). I2C: A System for the Indexing, Storage, and Retrieval of Medical Images by Content. Medical Informatics, 19(2), pp. 109-122.
11. Orphanoudakis, S., Tsiknakis, M., Chronaki, C., Kostomanolakis, S., Zikos, M., and Tsamardinos, Y. (1995). Development of an Integrated Image Management and Communication System on Crete. CAR'95, pp. 481-- 487, Berlin, June 21-24. Springer.
12. Orphanoudakis, S., Chronaki, C., and Vamvaka, D. (1996). I2Cnet: Content-Based Similarity Search in Geographically Distributed Repositories of Medical Images. Computerized Medical Imaging and Graphics, 20(4), pp. 193-207.
13. Sclaroff, S. World Wide Web image search engines. Tech. Report TR-95-16, Image and Video Computing Group, Somp. Sci., Boston University, Boston, MA 02215; 1995.
14. Smeulders, A., Worring, M., Santini, S., Gupta, A., and Jain, R. (2000). Content-Based Image Retrieval at the End of the Early Years. IEEE Trans. Pattern Analysis and Machine Intelligence, 22(12), pp. 1-32.
15. Swain, M. (1999). Searching for Multimedia on the World Wide Web. Proc. Int'l Conf. Multimedia Computing and Systems, pp. 11-32.
16. Swets, D.L., and Weng, J.J. (1995). Efficient Content-Based Image Retrieval using Automatic Feature Selection. In: Proc. International Symposium on Computer Vision, Coral Gables, Florida, pp. 85-90.
17. Swets, D., Weng, J. (1999). Hierarchical Discriminant Analysis for Image Retrieval. IEEE Trans. Pattern Analysis and Machine Intelligence, 21(5), pp. 386-401.
18. Weber, M., Welling, M., and Perona, P. (2000). Towards Automatic Discovery of Object Categories. Proc. Computer Vision and Pattern Recognition, pp. 101-108.
19. Wilson, D.R., and Martinez, T.R. (1997). Improved Heterogeneous Distance Functions. Journal of Artificial Intelligence Research, 6, 1-34.

Author Index